



**UNIVERSIDAD NACIONAL
PEDRO RUÍZ GALLO**



**FACULTAD DE CIENCIAS FISICAS Y
MATEMATICAS**

Escuela Profesional
Ing. Computación e Informática

**“Sistema de búsqueda heurística para la localización de las
principales oficinas de la Universidad Nacional Pedro Ruiz
Gallo utilizando el algoritmo A Star”**

**TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO EN COMPUTACIÓN E INFORMÁTICA**

INVESTIGADORES

Bach. Quiroz Dávila Jefferson Leandro

Bach. Ramírez López Kelvin Dante

ASESOR

Dr. Ing. Maquen Niño Gisella Luisa Elena

Lambayeque, 2019

“Sistema de búsqueda heurística para la localización de las principales oficinas de la Universidad Nacional Pedro Ruiz Gallo utilizando el algoritmo A Star”

Aprobado por:

Dr. Ing. Iván Eduardo Salvador Briceño

PRESIDENTE DE JURADO

M. Sc. Ing. Segundo Pedro Fiestas Rodríguez

SECRETARIO DE JURADO

Ing. Oscar Alex Serquén Yparraguirre

VOCAL DE JURADO

Dra. Maquen Niño Gisella Luisa Elena

ASESOR

DEDICATORIA

Dedico de manera especial a mis padres pues ello son los cimientos para la construcción de mi vida personal y profesional, sentaron en mí las bases de la responsabilidad y el deseo de superación, en ellos tengo el modelo en el cual quiero moldearme.

A mis amigos por darme siempre su apoyo, ánimos y palabras de aliento para continuar y alcanzar mis objetivos.

Ramírez López Kelvin Dante

Dedico de manera especial a mis padres y a mi familia pues ello son los que me formaron para ser la persona que soy, a mi esposa por su apoyo incondicional. A mi hija para que le sirva de ejemplo y siga los pasos de sus padres en ser una profesional.

Quiroz Dávila Jefferson Leandro

AGRADECIMIENTOS

Ramírez López Kelvin Dante:

Doy gracias a Dios por darme salud, amistad, amor de mi familia y permitirme tener una excelente formación académica en la Universidad Nacional Pedro Ruiz Gallo, gracias a mis padres por apoyarme cada día, a mi compañero de tesis Jefferson y a nuestra asesora por permitir hacer realidad esta meta de nuestra formación profesional.

Quiroz Dávila Jefferson Leandro:

Primero Dios, por darme la vida y por las oportunidades y enseñanzas que me da. A mis padres y familia por todo el apoyo incondicional que me han dado. A mi esposa Flor por apoyarme cuando lo he necesitado. A mi compañero de tesis que demostró ser un gran complemento para realizar este proyecto y a mi asesora por estar al pendiente con gran dedicación y esfuerzo para culminar nuestro grado en nuestra vida profesional.

RESUMEN

El presente trabajo de investigación manifiesta la siguiente interrogante, ¿Cómo se podrá facilitar la localización de las principales oficinas de la Universidad Nacional Pedro Ruiz Gallo?, Como respuesta al problema se planteó el siguiente objetivo que le dará solución:

Desarrollar un sistema de búsqueda heurística para la localización de las principales oficinas de la Universidad Nacional Pedro Ruiz Gallo utilizando el algoritmo A Star.

La idea de un sistema que facilite la localización de las principales oficinas de la Universidad Nacional Pedro Ruiz Gallo surge a partir de la necesidad de sus alumnos, docentes y allegados para encontrar con rapidez estos departamentos administrativos, puesto que la propia universidad les demanda realizar diversos trámites a lo largo de su trayectoria académica, en diversos departamentos.

El sistema de búsqueda heurística se desarrolló bajo dos conceptos, el primero es el de hallar rutas con sencillez gracias a la orientación de la heurística y el segundo es el de representar la universidad en un espacio virtual de modo que facilite al usuario el reconocimiento de su ubicación y trayectoria.

Es así, que el sistema de búsqueda heurística surge como solución a este problema de localización tan recurrente por los alumnos, docentes y allegados a la universidad Nacional Pedro Ruiz Gallo, evitándoles retrasos y pérdidas de tiempo en sus actividades diarias.

Palabras clave: localización, búsqueda, Heurística.

ABSTRACT

This research paper expresses the following question: How can the location of the main offices of the Pedro Ruiz Gallo National University be facilitated? In response to the problem, the following objective was set that will provide a solution:

Develop a heuristic search system for the location of the main offices of the National University Pedro Ruiz Gallo using the A Star algorithm.

The idea of a system that facilitates the location of the main offices of the National University Pedro Ruiz Gallo arises from the need of its students, teachers and relatives to quickly find these administrative departments, since the university itself demands them to perform various procedures throughout his academic career, in various departments.

The heuristic search system was developed under two concepts, the first is to find routes with simplicity thanks to the orientation of the heuristic and the second is to represent the university in a virtual space so that it facilitates the user to recognize their location and trajectory.

Thus, the heuristic search system emerges as a solution to this recurring location problem by students, teachers and those close to the Pedro Ruiz Gallo National University, avoiding delays and wasted time in their daily activities.

KEYWORDS: location, search, heuristics.

INTRODUCCIÓN

La presente tesis es una investigación que tiene por objetivo implementar un software utilizando el algoritmo A Star para localizar las principales oficinas de la Universidad Nacional Pedro Ruiz Gallo. La investigación se realizó por el interés de poner en práctica técnicas de búsqueda heurística que son aplicables en el desarrollo de diversas problemáticas, como es la de encontrar rutas en ambientes extensos. El desarrollo de esta investigación se realizó bajo el enfoque de la metodología Proceso Unificado Ágil; integrando el modelado del software mediante el lenguaje de modelado unificado. Este trabajo presenta los siguientes capítulos:

En el capítulo I se presenta el análisis del problema, que contiene la situación problemática, formulación del problema científico, la hipótesis, los objetivos, tipo de investigación, las variables, la sección de la metodología y algoritmo de búsqueda.

En el capítulo II se exhibe el diseño teórico, dentro del diseño teórico se abordan los programas, la base de datos utilizada, el lenguaje de programación, el algoritmo A Star y la metodología AUP, entre otros puntos más.

En el capítulo III se exhibe los métodos y materiales, que contiene el diseño de contratación de hipótesis, la población, la muestra, técnicas de recolección de datos y el instrumento de recolección de datos.

En el capítulo IV se presenta los resultados, que contiene el desarrollo de la metodología AUP en las siguientes fases: iniciación, elaboración, construcción y transición.

INDICE GENERAL

RESUMEN	5
INTRODUCCIÓN	7
CAPÍTULO I: Análisis del problema	19
1.1. Síntesis de la situación problemática	19
1.2. Formulación del problema.....	22
1.3. Hipótesis	22
1.4. Objetivos de la investigación.....	22
1.4.1. Objetivos generales	22
1.4.2. Objetivo específico	22
1.5. Operación de variables	23
1.5.1. Variable Independiente	23
1.5.2. Variable Dependiente.....	23
1.6. Selección de la metodología	24
1.7. Selección del algoritmo de búsqueda.....	26
1.8. Tipo de Investigación.....	27
CAPÍTULO II: Diseño Teórico	27
2.1. Teoría de grafos	28
2.2. Algoritmos para la ruta más corta en un Grafo.....	33
2.3. Técnica de búsqueda en árbol.....	34

2.4.	Estrategia de búsqueda informada	36
2.5.	Búsqueda A Star	36
2.6.	Geolocalización	38
2.7.	Sistema de Posicionamiento Global.....	39
2.8.	Antecedentes.....	40
2.9.	Programas	43
2.9.1.	Unity 3D	43
2.9.2.	Blender	44
2.9.3.	Gimp.....	44
2.9.4.	AutoCAD	45
2.9.5.	Xampp	45
2.9.6.	Sublime Text	45
2.10.	Lenguajes de programación	46
2.10.1.	C Sharp	46
2.10.2.	PHP.....	47
2.11.	Otros lenguajes.....	47
2.11.1.	HTML.....	47
2.11.2.	CSS	47
2.11.3.	JavaScript.....	48
2.12.	Base de Datos.....	49

2.12.1. MySQL	49
2.13. Proceso Unificado Ágil.....	49
CAPÍTULO III: Métodos y Materiales.....	55
3.1. Diseño de contratación de Hipótesis	55
3.2. Población y Muestra.....	56
3.2.1. Población.....	56
3.2.2. Muestra.....	56
3.3. Técnica de recolección de datos	56
3.4. Instrumento de recolección de datos	57
CAPÍTULO IV: Resultados	58
4.1. INICIACIÓN	58
4.1.1. Descripción del Sistema.....	58
4.1.2. Consideraciones sobre el sistema	59
4.1.3. Alcance del Proyecto	60
4.1.4. Programación de Actividades	75
4.1.5. Definición de Riesgos	76
4.1.6. Viabilidad del Proyecto.....	81
4.1.7. Asignación de funciones a Hardware y software	83
4.2. ELABORACIÓN	84
4.2.1. Identificación de la Arquitectura	84

4.2.2.	Vista lógica de la Arquitectura	86
4.2.3.	Validación de la Arquitectura.....	88
4.2.4.	Equipo de Trabajo.....	89
4.2.5.	Diagrama de Clases de Análisis	90
4.3.	CONSTRUCCIÓN	91
4.3.1.	Modelamiento del Sistema	91
4.3.2.	Construcción del Sistema	112
4.4.	TRANSICIÓN	120
4.4.1.	Pruebas del Sistema	120
4.4.2.	Pruebas con los usuarios	124
CONCLUSIONES		128
RECOMENDACIONES		128
BIBLIOGRAFIA		129
ANEXOS		133

INDICE DE TABLAS

<i>Tabla 1. Variable Independiente.....</i>	<i>23</i>
<i>Tabla 2. Variable Dependiente</i>	<i>23</i>
<i>Tabla 3. Cuadro comparativo entre metodologías</i>	<i>24</i>
<i>Tabla 4. Cuadro comparativo entre algoritmos de búsqueda.....</i>	<i>26</i>
<i>Tabla 5. Requerimientos Funcionales, Módulo de Administración</i>	<i>61</i>
<i>Tabla 6. Requerimientos Funcionales, Módulo de Usuario.....</i>	<i>61</i>
<i>Tabla 7. Requerimientos no Funcionales</i>	<i>62</i>
<i>Tabla 8. Especificación de Caso de Uso, Iniciar Sesión</i>	<i>65</i>
<i>Tabla 9. Especificación de Caso de Uso, Gestionar Usuario</i>	<i>66</i>
<i>Tabla 10. Especificación de Caso de Uso, Gestionar Nodo</i>	<i>67</i>
<i>Tabla 11. Especificación de Caso de Uso, Gestionar Destino.....</i>	<i>68</i>
<i>Tabla 12. Especificación de Caso de Uso, Gestionar Arista</i>	<i>69</i>
<i>Tabla 13. Especificación de caso de uso, Generar Grafo.....</i>	<i>70</i>
<i>Tabla 14. Especificación de Caso de Uso, Cargar Datos.....</i>	<i>70</i>
<i>Tabla 15. Especificación de Caso de Uso, Buscar Ruta.....</i>	<i>71</i>
<i>Tabla 16. Especificación de Caso de Uso, Consultar Destino</i>	<i>72</i>
<i>Tabla 17. Especificación de Caso de Uso, Consultar Origen</i>	<i>72</i>
<i>Tabla 18. Especificación de caso de uso, Generar Reporte Visitantes</i>	<i>73</i>
<i>Tabla 19. Matriz de Probabilidad e Impacto</i>	<i>77</i>
<i>Tabla 20. Identificación de Riesgos</i>	<i>77</i>
<i>Tabla 21. Respuesta ante Riesgos.....</i>	<i>79</i>
<i>Tabla 22. Viabilidad del Proyecto</i>	<i>81</i>

<i>Tabla 23. Costo de Personal</i>	<i>82</i>
<i>Tabla 24. Costo referencia del Proyecto.....</i>	<i>83</i>
<i>Tabla 25. Roles y Responsabilidades</i>	<i>89</i>
<i>Tabla 26. Catálogo de Pruebas</i>	<i>121</i>
<i>Tabla 27. Encuesta N° 1, Respuestas de las personas encuestadas</i>	<i>136</i>
<i>Tabla 28. Tiempo Experimental.....</i>	<i>139</i>
<i>Tabla 29. Tiempo Control</i>	<i>139</i>
<i>Tabla 30. Plantilla de prueba de caja negra.....</i>	<i>142</i>
<i>Tabla 31. Resultados de las pruebas de caja negra</i>	<i>143</i>
<i>Tabla 32. Encuesta N° 2, Respuestas de las personas encuestadas</i>	<i>215</i>

INDICE DE FIGURAS

Figura 1: Visualización Estadística de la Encuesta N° 1 Pregunta 1	20
Figura 2: Grafico Comparativo de Tiempo control y Tiempo experimental promedio	21
Figura 3. Modelo de Grafo	28
Figura 4: Grafo orientado	29
Figura 5: Grafo no orientado	30
Figura 6: Lista de incidencia	30
Figura 7: Lista de adyacencia	31
Figura 8: Matriz de incidencia	31
Figura 9: Matriz de adyacencia.....	32
Figura 10: Grafo representado el mapa de Rumanía.....	34
Figura 11: Árboles de búsqueda parciales para encontrar una ruta desde Arad hasta Bucarest....	35
Figura 12: Árbol de búsqueda A* para Bucarest	38
Figura 13. Ciclo de vida AUP.....	50
Figura 14. Diagrama de caso de uso del sistema	64
Figura 15. Diagrama WBS	74
Figura 16. Diagrama de Gantt	75
Figura 17. Arquitectura Cliente – Servidor	84
Figura 18. Vista lógica de la arquitectura.....	86
Figura 19. Diagrama de Clases de Análisis	90
Figura 20. Diagrama de Paquetes	91
Figura 21. Diagrama de Clases de Diseño	93
Figura 22. Diagrama de Componentes.....	94

Figura 23. Diagrama de Despliegue.....	94
Figura 24. Diagrama de Secuencia - Iniciar Sesión.....	95
Figura 25. Diagrama de Secuencia - Gestionar Usuario	96
Figura 26. Diagrama de Secuencia - Gestionar Nodo.....	97
Figura 27. Diagrama de Secuencia - Gestionar Arista.....	98
Figura 28. Diagrama de Secuencia – Gestionar Destino.....	99
Figura 29. Diagrama de Secuencia – Generar Grafo	100
Figura 30. Diagrama de Secuencia – Generar Reporte Visitantes.....	101
Figura 31. Diagrama de Secuencia - Cargar Datos.....	102
Figura 32. Diagrama de Secuencia - Consultar Origen.....	103
Figura 33. Diagrama de Secuencia - Consultar Destino	104
Figura 34. Diagrama de Secuencia - Buscar Ruta	105
Figura 35. Diagrama de Base de Datos	106
Figura 36. Interfaz Gráfica – Cargar Datos	107
Figura 37. Interfaz Gráfica - Buscar Ruta	107
Figura 38. Interfaz Gráfica - Consultar Origen	107
Figura 39. Interfaz Gráfica - Consultar Destino	108
Figura 40. Interfaz Gráfica - Iniciar Sesión.....	108
Figura 41. Interfaz Gráfica - Gestionar Usuario	109
Figura 42. Interfaz Gráfica- Gestionar Nodo.....	109
Figura 43. Interfaz Gráfica - Gestionar Destino	110
Figura 44. Interfaz Gráfica - Consultar Nodo.....	110
Figura 45. Interfaz Gráfica- Generar grafo.....	111

Figura 46. Interfaz Gráfica- Generar Reporte Visitantes	111
Figura 47. Construcción de la Base de Datos	117
Figura 48. Estructura de la Aplicación Web.....	118
Figura 49. Estructura de la Interfaz de Programación de Aplicaciones	118
Figura 50. Estructura de la Aplicación Móvil	119
Figura 51: Visualización Estadística de la Encuesta N° 2 Pregunta 1	124
Figura 52: Visualización Estadística de la Encuesta N° 2 Pregunta 2	125
Figura 53: Visualización Estadística de la Encuesta N° 2 Pregunta 3	125
Figura 54: Visualización Estadística de la Encuesta N° 2 Pregunta 4	126
Figura 55: Visualización Estadística de la Encuesta N° 2 Pregunta 5	126
Figura 56: Visualización Estadística de la Encuesta N° 2 Pregunta 6	127
Figura 57: Visualización Estadística de la Encuesta N° 2 Pregunta 7	127
Figura 58: Plano de La Universidad Nacional Pedro Ruíz Gallo	140
Figura 59: Grafo que representa la Universidad Nacional Pedro Ruíz Gallo	141
Figura 60: Login.jpg.....	170
Figura 61: Sprite 2D Buscar	185
Figura 62: Sprite 2D Ruta.....	186
Figura 63: Sprite 2D Ubicacion	186
Figura 64: Sprite 2D Desplazar	187
Figura 65: Sprite 2D Seleccionar	187
Figura 66: Sprite 2D Salir.....	188
Figura 67: Sprite 2D UNPRG.....	188
Figura 68: Sprite 2D Icono	189

Figura 69: Mesh de las aulas de Derecho, Objeto 3D.....	189
Figura 70: Mesh del Pabellón de Aulas, Objeto 3D	190
Figura 71: Mesh de las aulas block “A”, Objeto 3D.....	190
Figura 72: Mesh de FICSA, Objeto 3D.....	191
Figura 73: Mesh del laboratorio de ing. Civil, Objeto 3D	191
Figura 74: Mesh del Coliseo, Objeto 3D.....	192
Figura 75: Mesh del Laboratorio de Arquitectura, Objeto 3D	192
Figura 76: Mesh del laboratorio de Ciencias Biologicas, Objeto 3D	193
Figura 77: Mesh de Recursos Hidricos, Objeto 3D	193
Figura 78: Mesh del Laboratorio de Electronica, Objeto 3D	194
Figura 79: Mesh del pabello de aulas de FACEAC, Objeto 3D.....	194
Figura 80: Mesh de las aulas de Veterinaria, Objeto 3D	195
Figura 81: Mesh de las aulas de FACHSE block “1”, Objeto 3D	195
Figura 82: Mesh de FACEAC, Objeto 3D	196
Figura 83: Mesh de las aulas de FACHSE block “2”., Objeto 3D	196
Figura 84: Mesh de FIZ, Objeto 3D.....	197
Figura 85: Mesh del Auditorio, Objeto 3D	197
Figura 86: Mesh de la Biblioteca, Objeto 3D.....	198
Figura 87: Mesh de la Ofcinica Central de Asuntos Academicos, Objeto 3D	198
Figura 88: Mesh de la Puerta Sur, Objeto 3D.....	199
Figura 89: Mesh de la Banda de Concierto, Objeto 3D	199
Figura 90: Mesh de las aulas de Medicina Humana, Objeto 3D	200
Figura 91: Mesh de la Facultad de Enfermeria, Objeto 3D.....	200

Figura 92: Mesh del Laboratorio de Biología y Microbiología, Objeto 3D.....	201
Figura 93: Mesh del Laboratorio de Fitopatología, Objeto 3D	201
Figura 94: Mesh del Auditorio Principal, Objeto 3D.....	202
Figura 95: Mesh de la Barracas 1, Objeto 3D	202
Figura 96: Mesh de la Facultad de Medicina Humana, Objeto 3D	203
Figura 97: Mesh de la Barracas 2, Objeto 3D	203
Figura 98: Mesh de Grados y Titulos, Objeto 3D.....	204
Figura 99: Mesh del Vivero, Objeto 3D.....	204
Figura 100: Mesh de las aulas block “B”, Objeto 3D	205
Figura 101: Mesh de las aulas block “C”, Objeto 3D	205
Figura 102: Mesh de las Biblioteca Principal, Objeto 3D.....	206
Figura 103: Mesh de la Red Telemática, Objeto 3D.....	206
Figura 104: Mesh del Laboratorio de Semillas, Objeto 3D	207
Figura 105: Mesh de la Oficina de Planillas, Objeto 3D	207
Figura 106: Mesh de la Oficina Central de Investigación, Objeto 3D.....	208
Figura 107: Mesh del Laboratorio de Mecánica, Objeto 3D	208
Figura 108: Mesh de FIME, Objeto 3D	209
Figura 109: Mesh del Centro de Producción y Servicios Electromecánicos, Objeto 3D	209
Figura 110: Mesh de las aulas de Mecánica, Objeto 3D	210
Figura 111: Mesh de la Plaza del Saber, Objeto 3D	210
Figura 112: Mesh de la Puerta Principal, Objeto 3D	211
Figura 113: Aplicación móvil, Vista 2D	212
Figura 114: Aplicación móvil, Vista 3D	212

CAPÍTULO I: Análisis del problema

1.1. Síntesis de la situación problemática

En el año 2016, la Pontificia Universidad Católica del Perú (PUCP) manifestó la siguiente problemática: Debido a la gran extensión que ocupa una universidad como esta y a la cantidad de edificios construidos en ella, una persona que no pertenece o que recién se incorpora a la comunidad universitaria no conoce la ubicación de lugares importantes dentro del campus universitario como las unidades académicas, facultades o bibliotecas; incluso las personas que sí pertenecen a la comunidad universitaria pueden no conocer la ubicación de lugares menos visitados como las direcciones administrativas, centros e institutos.

La solución a este problema fue propuesta por la Dirección Informática Académica (DIA) de la PUCP el cual consistía en desarrollar una aplicación que ayude a sus usuarios a identificar lugares y eventos del campus universitario, que los oriente para llegar a su destino y que les muestre información detallada de los mismos; al cual llamo “Descubre PUCP”. (Carrión, 2016, p.2)

La Universidad Nacional Pedro Ruiz Gallo desde 1970 que fue fundada hasta la actualidad ha estado en constante desarrollo tanto institucionalmente como en su infraestructura, contando hasta el día de hoy con 14 facultades y 30 escuelas profesionales, las que albergan bianualmente una gran afluencia de alumnos. Los cuales en algún punto presentan el mismo problema en común, la dificultad en la localización de alguna oficina del campus universitario al realizar un trámite universitario.

Para medir el nivel de esta problemática en septiembre del 2018 se llevó a cabo un muestreo no probabilístico de tipo convencional, estableciéndose una cuota de 63 personas

por conveniencia de los autores para realizar un cuestionario, donde se les ha hecho preguntas relacionadas con la localización de determinadas oficinas; según el gráfico de la siguiente Figura 1 se obtuvo que un 68% de los alumnos ha tenido dificultad en localizar las oficinas de la universidad por lo menos ocasionalmente (Ver Anexo 1).

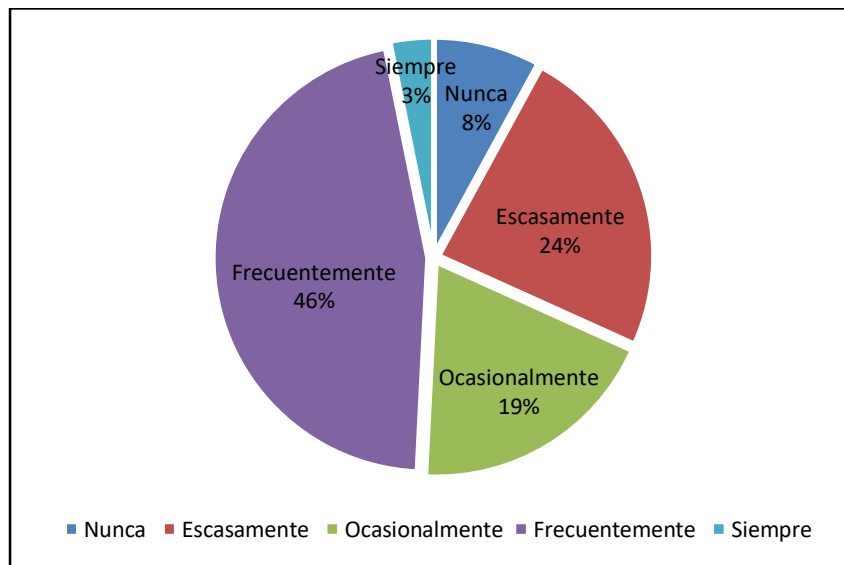


Figura 1: Visualización Estadística de la Encuesta N° 1 Pregunta 1
Fuente: Elaboración Propia

El desconocimiento de los alumnos en la ubicación de las principales oficinas de la universidad conlleva a que realicen un recorrido excesivo para ubicarlas en el campus, o que soliciten información a otros miembros de la universidad. Aunque dicha información se hace de manera verbal, lo que significa que esta información es empírica. Estas dificultades contribuyen a que los alumnos pierdan mucho de su tiempo y que parte de sus actividades diarias no sean culminadas o queden inconclusas.

Para determinar el tiempo perdido por los estudiantes al buscar una oficina de la universidad Nacional Pedro Ruiz Gallo se ha realizado un experimento, tomando una muestra de 10 individuos, los cuales desconocen el campus.

El cual consistía en ubicar la Facultad de Ciencias Físicas y Matemáticas desde la puerta principal de la universidad. Recaudando los datos del tiempo que se ha demorado cada participante, hemos obtenido los siguientes resultados:

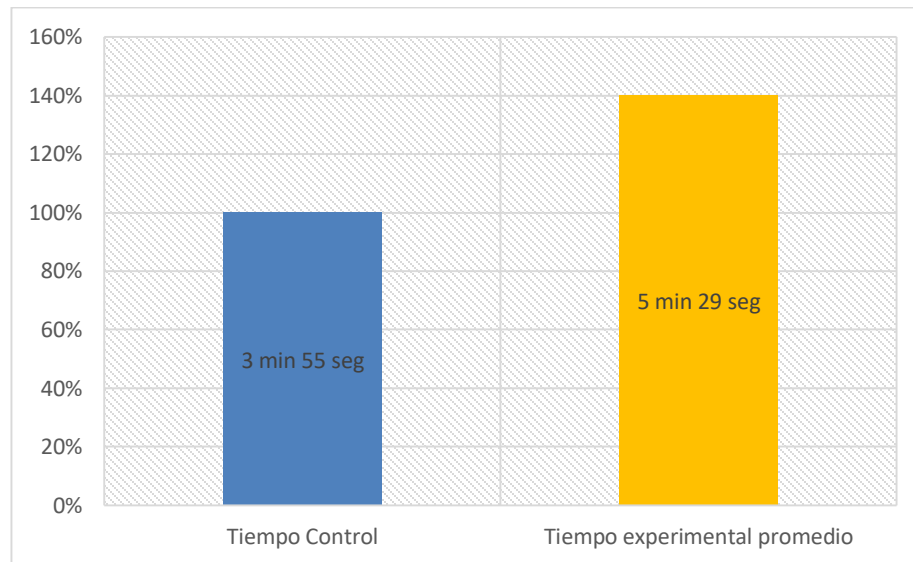


Figura 2: Grafico Comparativo de Tiempo control y Tiempo experimental promedio
Fuente: Elaboración Propia

Los participantes demostraron una tardanza, sin excepción. Comparado con el desplazamiento del investigador, que sí conocen la universidad, se estableció un promedio de tardanza que tienen los participantes, el cual es el 40% de tiempo excedido (ver Anexo 2), debido a la falta de conocimiento y orientación verbal que solicitaban al público, que transitaban por la universidad.

Cabe mencionar como experiencia propia de los investigadores, que han sido alumnos de la universidad, han afrontado este tipo de situaciones, para realizar distintos trámites, los cuales no sólo les exigía conocer las oficinas correspondientes a su facultad, sino también de otras áreas de la universidad. Durante la carrera como al finalizarla.

Un medio para resolver este problema es realizar una herramienta virtual para el óptimo desplazamiento de los alumnos en la Universidad Nacional Pedro Ruiz Gallo de

Lambayeque. Es un recurso conveniente debido a la gran masa de personas que transitan la universidad a diario, por diversos motivos, sobre todo los que competen al ámbito administrativo.

1.2. Formulación del problema

¿Cómo se podrá facilitar la localización de las principales oficinas de la Universidad Nacional Pedro Ruiz Gallo?

1.3. Hipótesis

El desarrollo de un sistema de búsqueda heurístico facilitará la localización de las principales oficinas de la Universidad Nacional Pedro Ruiz Gallo.

1.4. Objetivos de la investigación

1.4.1. Objetivos generales

Desarrollar un sistema de búsqueda heurística para la localización de las principales oficinas de la Universidad Nacional Pedro Ruiz Gallo utilizando el algoritmo A Star.

1.4.2. Objetivo específico

- Desarrollar el sistema informático bajo la metodología Proceso Unificado Ágil.
- Diseñar la estructura de la Universidad Nacional Pedro Ruiz Gallo en 3D utilizando el programa informático Blender.
- Seleccionar el algoritmo de búsqueda más adecuado para la investigación.
- Establecer los nodos que constituyen el grafo de las vías transitables de la Universidad Nacional Pedro Ruiz Gallo.

- Desarrollar la Base de Datos en Mysql para la administración del grafo y de los usuarios del sistema de búsqueda heurística.
- Efectuar pruebas de los módulos del sistema.

1.5. Operación de variables

1.5.1. Variable Independiente

Tabla 1.
Variable Independiente

Variables Independiente	Definición	Dimensiones	Indicadores
Sistema de búsqueda heurístico	Sistema que dispone de información del estado objetivo, lo que permite explorar primero los caminos más prometedores.	Datos	Coordenadas de entrada y coordenadas de salida.
		Nodos	Número de nodos utilizados en la ruta.

Fuente: Elaboración propia

1.5.2. Variable Dependiente

Tabla 2.
Variable Dependiente

Variables Dependiente	Definición	Dimensiones	Indicadores
Proceso de localización de las principales oficinas.	Delimitación y ubicación de un lugar. Para que esta sea trazable y comunicable.	Distancia	Número de metros hallados por el sistema, para la ruta encontrada.

Fuente: Elaboración propia

1.6. Selección de la metodología

A continuación, en la Tabla 3 se presentan cuatro propuestas de metodologías para su evaluación y elección, la metodología seleccionada será utilizada en el presente proyecto de investigación para luego exponer justificadamente respecto a su elección:

Tabla 3.
Cuadro comparativo entre metodologías

<i>Criterios</i>	<i>Metodología de desarrollo de software</i>			
	<i>RUP</i>	<i>AUP</i>	<i>XP</i>	<i>SCRUM</i>
Descripción Breve	Constituye la metodología estándar más utilizada, centrada en la arquitectura del sistema. Se caracteriza por ocupar el modelo iterativo e incremental	Es una versión simplificada del proceso Unificado Rational. Usa técnicas ágiles y componentes que aún se mantienen en RUP	Es el más destacado de las metodologías ágiles. Modelo en el que se define un plan para el desarrollo y liberación de software.	Es una metodología ágil para trabajar en equipo a partir de iteraciones, por lo que su objetivo es controlar y planificar grandes proyectos
Tipo de FrameWork	Análisis, diseño, implementación y documentación de sistemas orientados a objetos	Basado en los procesos del desarrollo de software y documentación flexibles en sistemas orientados a objetos	Basado en la adaptabilidad, mayor flexibilidad, dinámico y funcional.	Gestión y desarrollo de software, basado en un proceso iterativo e incremental.
Tipo de revisión	En cada fase de la metodología se realiza una o más iteraciones, perfeccionando así los objetivos, si no se termina una fase no se continúa con la siguiente.	El desarrollo de las fases de la metodología es lineal, y se define el número de iteraciones en cada una para mejorar los resultados.	Se debe integrar como mínimo una vez al día, y realizar las pruebas sobre la totalidad del proceso.	Se necesita de una revisión diaria, se describen las siguientes 3 cuestiones: <ul style="list-style-type: none"> • Trabajo realizado el día anterior. • Trabajo previsto a realizar • Cosas que puede realizar o impedimento

Fuente: (Fernando Caceres, 2017, Metodologías de desarrollo de software)

Continuación de la Tabla 3, Cuadro comparativo entre metodologías

Objetivos	Orientado a objetos que establece las bases, plantillas y ejemplos para todos los aspectos y fases de desarrollo de software.	Especialista en la gestión de riesgos, propone aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas	Basada en dar prioridad a trabajos con resultado directo: <ul style="list-style-type: none"> • Satisfacción cliente • Trabajo en grupo • Actuar sobre variables: Coste, tiempo, calidad y alcance. 	Indicado para proyectos en entornos complejos: <ul style="list-style-type: none"> • Obtener resultados pronto • Requisitos cambiantes • Innovación y competitividad fundamentales.
Tipo de desarrollo	Proceso iterativo incremental por fases: <ul style="list-style-type: none"> • Inicio • Elaboración • Construcción • Transición 	Proceso iterativo incremental por fases: <ul style="list-style-type: none"> • Inicio • Elaboración • Construcción • Transición 	Liviana y adaptabilidad. Desarrollo por fases: <ul style="list-style-type: none"> • Planificación del proyecto. • Diseño • Codificación • Pruebas 	<ul style="list-style-type: none"> • Desarrollo simple, que requiere trabajo duro. • Control de forma empírica y adaptable a la evolución del proyecto.
Tipo de Proyecto	Recomendado para grandes, a largo plazo, a nivel de empresa con proyectos a medio y alta complejidad.	Utilizado en medianos y pequeños proyectos, a corto plazo, para evitar la documentación tediosa en los equipos de trabajo.	Se usa principalmente para proyectos pequeños a raíz de la desventaja de no precisar el costo del proyecto.	Recomendado para las mejoras rápidas y organizaciones que no dependen de una fecha límite.

Fuente: (Fernando Caceres, 2017, Metodologías de desarrollo de software)

La metodología de desarrollo seleccionada para el presente proyecto es Proceso Unificado Ágil por las razones expuestas a continuación:

- Puesto que el proyecto de investigación se está realizando sólo con fines académicos, existe un escaso costo de inversión y un tiempo corto para su desarrollo, por lo cual es propicio optar por una metodología ágil que permita cumplir con las cuotas de tiempo, pero que también permita documentar los procesos como propone el enfoque AUP.
- La metodología AUP ofrece un amplio marco de buenas prácticas en la fase de construcción de software en búsqueda de la optimización promoviendo medidas como la ejecución de pruebas en paralelo con la programación así como el manejo de unidades de prueba.
- Finalmente por tratarse de un equipo de proyecto conformado únicamente por dos tesista como responsable de las labores de análisis, diseño e implementación del software, el escenario resulta propicio optar la metodología AUP considerando su aplicación en entornos organizacionales no masivos o en equipos con una estructura jerárquica reducida.

1.7. Selección del algoritmo de búsqueda

En la Tabla 4 se proponen tres algoritmos de búsqueda reconocidos por ser utilizados en diversos problemas de caminos en grafos. Los algoritmos propuestos serán comparados y evaluados para seleccionar la opción que mejor se ajusta a los parámetros del proyecto.

Tabla 4.
Cuadro comparativo entre algoritmos de búsqueda

<i>Algoritmos de Búsqueda</i>

Criterios	Primero en anchura	Primero en profundidad	A Star
Entorno	Grafo bidireccional, unidireccional y mixto.	Grafo bidireccional, unidireccional y mixto.	Grafo bidireccional, unidireccional y mixto.
Técnica de búsqueda	Búsqueda –árboles	Búsqueda -árboles	Búsqueda -árboles
Tipo de búsqueda	Búsqueda no informada	Búsqueda no informada	Búsqueda informada
Orden de expansión	listas FIFO (cola)	listas LIFO (pila)	Heurística
Optimo	Cuando todos las acciones tiene el mismo coste	Cuando la profundidad de los nodos es limitada	Cuando se tiene una buena calidad en la heurística
Efectividad en condiciones optimas	Alta	Media	Alta
Procesamiento	Alto	Alto	Bajo
Consumo de Memoria	Alto	Bajo	Bajo
Complejidad	Baja	Baja	Alta

Fuente: Elaboración propia

El algoritmo de búsqueda seleccionado para el presente proyecto es el A Star, por los argumentos expuestos a continuación:

- Utiliza pocos recursos de memoria.
- Se desenvuelve correctamente sin importar la distribución del grafo.
- Se desenvuelve correctamente sin importar la cantidad de nodos en el grafo.
- Tiene pocas condiciones para su implementación.
- El resultado de la búsqueda es confiable.
- Se evalúan menos nodos que otros algoritmos gracias a la heurística.

1.8. Tipo de Investigación

Tecnológica – Formal

CAPÍTULO II: Diseño Teórico

2.1. Teoría de grafos

En matemáticas y en ciencias de la computación, la teoría de grafos (también llamada teoría de las gráficas) estudia las propiedades de los grafos. Un grafo es un conjunto, no vacío, de objetos llamados vértices (o nodos) y una selección de pares de vértices, llamados aristas que pueden ser orientados o no. Típicamente, un grafo se representa mediante una serie de puntos (los vértices) conectados por líneas (las aristas).

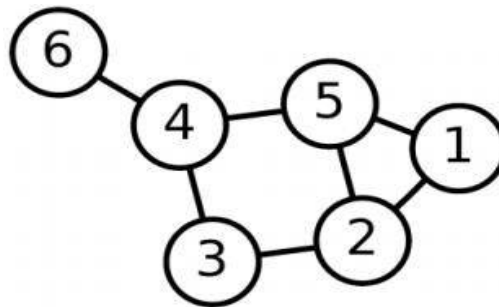


Figura 3. Modelo de Grafo
Fuente: (Universidad de Pamplona, 2012, Teoría de grafo)

Vértice:

Los vértices constituyen uno de los dos elementos que forman un grafo. Como ocurre con el resto de las ramas de las matemáticas, a la Teoría de Grafos no le interesa saber qué son los vértices. Diferentes situaciones en las que pueden identificarse objetos y relaciones que satisfagan la definición de grafo pueden verse como grafos y así aplicar la Teoría de Grafos en ellos.

Arista:

Una arista es la relación o enlace entre dos vértices que constituyen un grafo, estas se pueden clasificar de dos formas:

Aristas dirigidas y no dirigidas:

En algunos casos es necesario asignar un sentido a las aristas, por ejemplo, si se quiere representar la red de las calles de una ciudad con sus direcciones únicas. El conjunto de aristas será ahora un subconjunto de todos los posibles pares ordenados de vértices, con $(a, b) \neq (b, a)$. Los grafos que contienen aristas dirigidas se denominan grafos orientados, como el siguiente:

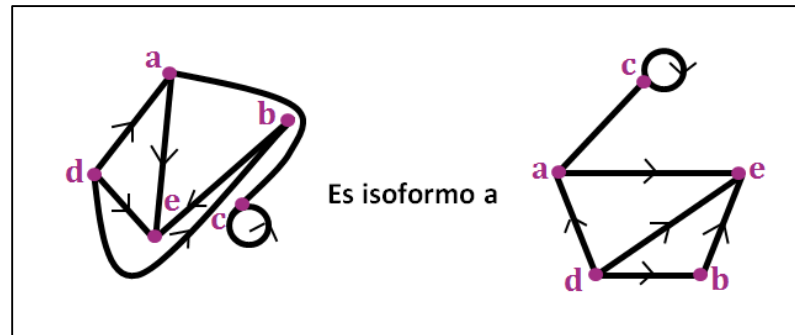


Figura 4: Grafo orientado
Fuente: (Universidad de Pamplona, 2012, *Teoría de grafos*)

Las aristas no orientadas se consideran bidireccionales para efectos prácticos (equivale a decir que existen dos aristas orientadas entre los nodos, cada una en un sentido).

Grafo:

Un grafo es una pareja de conjuntos $G = (V, A)$, donde V es el conjunto de vértices, y A es el conjunto de aristas, este último es un conjunto de pares de la forma (u, v) tal que $u, v \in V$. Para simplificar, notaremos la arista (a, b) como ab . En teoría de grafos, sólo queda lo esencial del dibujo: la forma de las aristas no son relevantes, sólo importa a qué vértices están unidas. La posición de los vértices tampoco importa, y se puede variar para obtener un dibujo más claro.

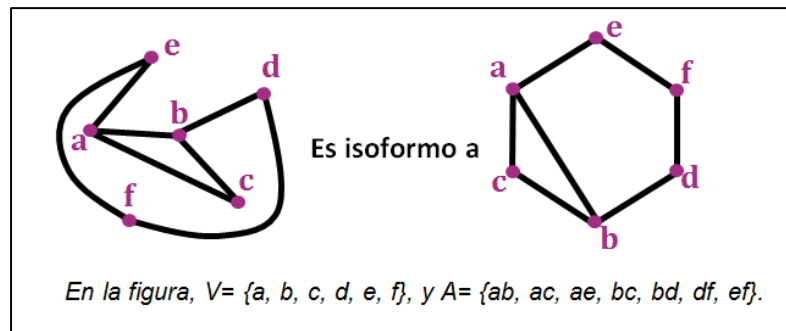


Figura 5: Grafo no orientado
Fuente: (Universidad de Pamplona, 2012, Teoría de grafos)

Estructuras de datos en la representación de grafos:

La estructura de datos usada depende de las características del grafo y el algoritmo usado para manipularlo. Entre las estructuras más sencillas y usadas se encuentran las listas y las matrices.

- Estructura de lista:
 - *Lista de incidencia:* Las aristas son representadas con un vector de pares donde cada par representa una de las aristas.

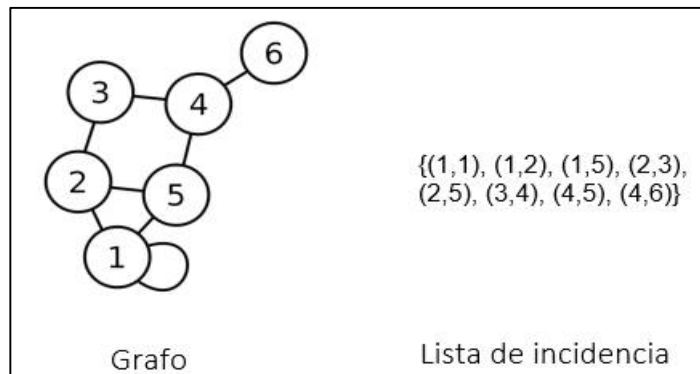


Figura 6: Lista de incidencia
Fuente: Elaboración Propia

- *Lista de adyacencia:* Cada vértice tiene una lista de vértices los cuales son adyacentes a él. Esto causa redundancia en un grafo no dirigido, pero las búsquedas son más rápidas, al costo de almacenamiento extra.

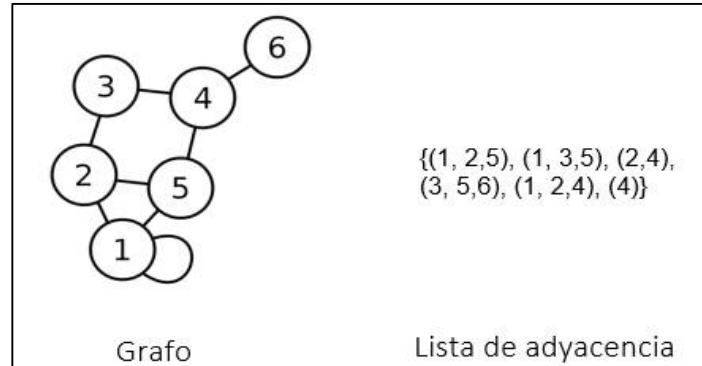


Figura 7: Lista de adyacencia

Fuente: Elaboración Propia

- Estructura de Matriz:
 - *Matriz de incidencia:* El grafo está representado por una matriz de A (aristas) y por V (vértices), donde [arista, vértice] contiene la información de la arista (1 - conectado, 0 - no conectado).

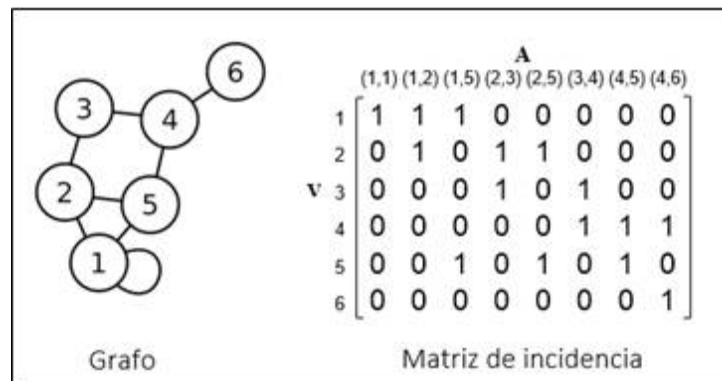


Figura 8: Matriz de incidencia

Fuente: Elaboración Propia

- *Matriz de adyacencia:* El grafo está representado por una matriz cuadrada M de tamaño n^2 , donde n es el número de vértices. Si hay una arista entre un vértice x y un vértice y , entonces el elemento $m_{x,y}$ es 1, de lo contrario, es 0.

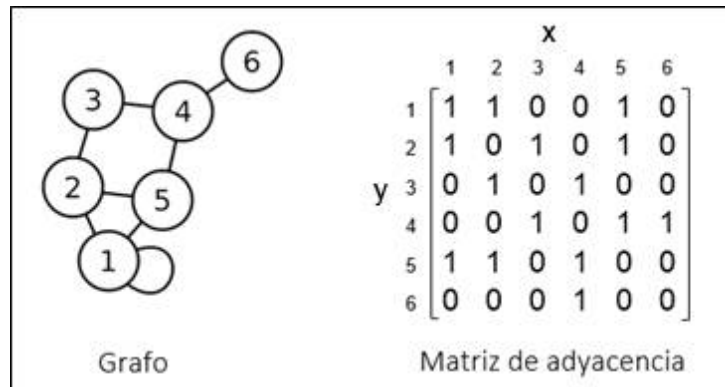


Figura 9: Matriz de adyacencia
Fuente: Elaboración Propia

Aplicaciones:

Gracias a la teoría de grafos se pueden resolver diversos problemas en diferentes áreas como por ejemplo:

- Dentro de las ciencias sociales para cuantificar relaciones complejas.
- En la electrónica para resolver problemas asociados a la síntesis de circuitos integrados, contadores o sistemas de aperturas.
- En la computación para proyectar redes de ordenadores.
- En el turismo para planificar trayectos y obtener rutas optimo.
- En la biología para ayudar al estudio de la migración de animales donde los vértices representan los hábitats y las aristas representa los senderos de los animales.

(Universidad de Pamplona, 2012)

2.2. Algoritmos para la ruta más corta en un Grafo.

Existen numerosos problemas que se pueden formular en términos de grafos. Ejemplo de ello es la planificación de las tareas que completan un proyecto, encontrar las rutas de menor longitud entre dos puntos geográficos, calcular el camino más rápido en un transporte, determinar el flujo máximo que puede llegar desde un punto A a un punto B.

La resolución de estos problemas requiere examinar todos los nodos o todas las aristas del grafo que representa al problema; sin embargo, existen ocasiones en que la estructura del problema es tal que sólo se necesitan visitar algunos de los nodos o bien algunas de las aristas. Los algoritmos imponen implícitamente un orden en estos recorridos: visitar el nodo más próximo o las aristas más cortas; y así sucesivamente; otros algoritmos no requieren ningún orden concreto en el recorrido.

Cuando se trabaja con grafos dirigidos etiquetados o ponderados con factores de peso no negativos, es frecuente buscar el camino más corto entre dos vértices dados; es decir, el camino que nos permita llegar desde un vértice origen a un vértice destino recorriendo la menor distancia o con el menor costo.

Los algoritmos más usados para este fin son:

- Algoritmo de búsqueda en anchura (BFS)
- Algoritmo de búsqueda en profundidad (DFS)
- Algoritmo de búsqueda A*
- Algoritmo del vecino más cercano
- Ordenación topológica de un grafo
- Algoritmo de Dijkstra
- Algoritmo de Bellman-Ford
- Algoritmo de Prim

- Algoritmo de Ford-Fulkerson
- Algoritmo de Kruskal
- Algoritmo de Floyd-Warshall

(Universidad Don Bosco, 2017, p.2)

2.3. Técnica de búsqueda en árbol

La técnica de búsqueda en árbol tiene como propósito encontrar un estado objetivo o destino en una colección de nodos vinculados con una relación de jerarquía. Esta técnica utiliza un árbol generado por el estado inicial y una función llamada sucesor (acción que permite alcanzar otros estados desde un estado origen), definiendo así todos los estados alcanzables desde el estado inicial, también llamado espacio de estados.

A continuación se utilizará un ejemplo para explicar con mayor detalle esta técnica:

La Figura 10 muestra un grafo compuesto por las ciudades de Rumanía y la Figura 11 un ejemplo de algunas de las expansiones en el árbol de búsqueda con un estado inicial en Arad.

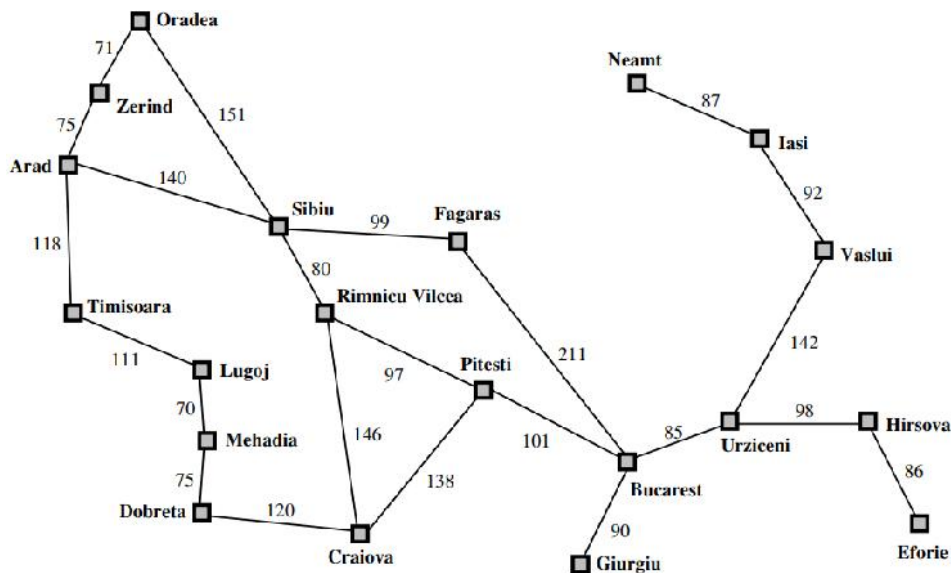


Figura 10: Grafo representado el mapa de Rumanía
Fuente: (Stuart J. Russell, & Peter Norvig, 2008, Inteligencia Artificial un Enfoque Moderno)

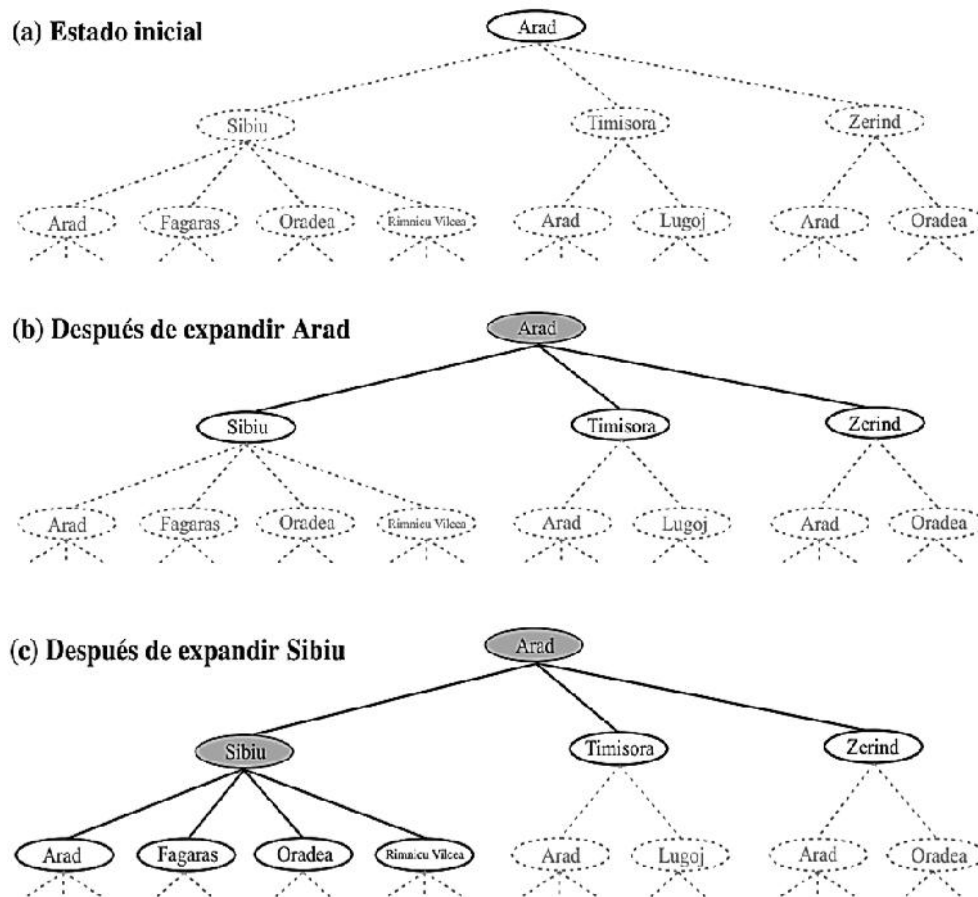


Figura 11: Árboles de búsqueda parciales para encontrar una ruta desde Arad hasta Bucarest
Fuente: (Stuart J. Russell, & Peter Norvig, 2008, Inteligencia Artificial un Enfoque Moderno)

Descripción Figura 11:

“Expansión del árbol de búsqueda parcial para encontrar una ruta”.

- Pasó 1: Se sombrea los nodos que se han expandido.
- Pasó 2: Los nodos que se han generado pero todavía no se han expandido se rodean en negrita
- Pasó 3: Los nodos que todavía no se han generado se marcan con líneas discontinuas.

La raíz del árbol de búsqueda es el nodo de búsqueda que corresponde al estado inicial, En (Arad). El primer paso es comprobar si este es un estado objetivo. Claramente es que no, como no estamos en un estado objetivo, tenemos que considerar otros estados. Esto se hace expandiendo el estado actual; es decir aplicando la función sucesor al estado actual y generar así un nuevo conjunto de estados. En este caso, conseguimos tres nuevos estados:

En (Sibiu), en (Timisoara) y en (Zerind). Ahora debemos escoger cuál de estas tres posibilidades podemos considerar. Esto es la esencia de la búsqueda, llevamos a cabo una opción y dejamos de lado las demás para más tarde, en caso de que la primera opción no conduzca a una solución. Supongamos que primero elegimos Sibiu. Comprobamos si es un estado objetivo (que no lo es) y entonces expandimos para conseguir En (Arad), en (Fagaras), en (Oradea) y en (RimnicuVilcea). Entonces podemos escoger cualquiera de estas cuatro o volver atrás y escoger Timisoara o Zerind. Continuamos escogiendo, comprobando y expandiendo hasta que se encuentra una solución o no existan más estados para expandir. (Russell & Norvig, 2008, p. 78)

2.4. Estrategia de búsqueda informada

La estrategia de la búsqueda informada también llamada búsqueda heurística utiliza el conocimiento específico del problema para alcanzar el objetivo de una manera más eficiente que una estrategia no informada, la idea es ser capaces de medir la “calidad” de un estado, eso nos permitirá dirigir la búsqueda por mejores estados, acercándonos más al objetivo. Las estrategias de búsqueda informada son mucho más eficientes que las no informadas ya que tienen conocimiento del objetivo y puede impedir a los algoritmos cometer errores en la oscuridad y caer en bucles infinitos. (Russell & Norvig, 2008, p. 108)

2.5. Búsqueda A Star

La Búsqueda A estrella es un algoritmo de búsqueda informada para grafos que encuentra el camino de menor coste entre un nodo inicial y un nodo meta. El procedimiento para el desarrollo de este algoritmo es el siguiente:

Evalúa los nodos combinando $g(n)$, el coste para alcanzar el nodo, y $h(n)$, el coste de ir al nodo objetivo:

$$f(n) = g(n) + h(n)$$

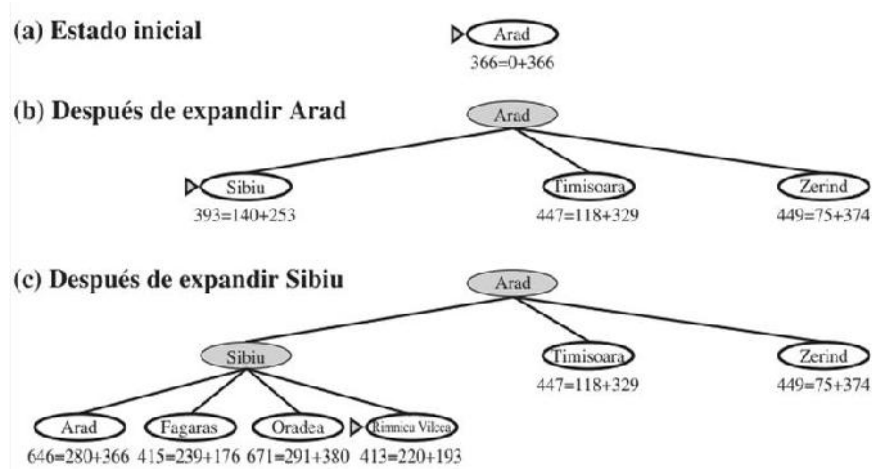
Ya que la $g(n)$ nos da el coste del camino desde el nodo inicio al nodo n , y la $h(n)$ el coste estimado del camino más barato desde n al objetivo, tenemos:

$$f(n) = \text{coste más barato estimado de la solución a través de } n.$$

Así, si tratamos de encontrar la solución más barata, es razonable intentar primero el nodo con el valor más bajo de $g(n) + h(n)$. Resulta que esta estrategia es más que razonable: con tal de que la función heurística $h(n)$ satisfaga ciertas condiciones, la búsqueda A* es tanto completa como óptima.

La optimización de A* es sencilla de analizar si se usa con la BÚSQUEDA-ÁRBOLES. En este caso, A* es óptima si $h(n)$ es una heurística admisible, es decir, con tal de que la $h(n)$ nunca sobrestime el coste de alcanzar el objetivo. Las heurísticas admisibles son por naturaleza optimistas, porque piensan que el coste de resolver el problema es menor que el que es en realidad. Ya que $g(n)$ es el coste exacto para alcanzar n , tenemos como consecuencia inmediata que la $f(n)$ nunca sobrestima el coste verdadero de una solución a través de n . (Russell & Norvig, 2008, p. 111)

La siguiente figura muestra un ejemplo de las etapas de la búsqueda A* para encontrar la ciudad de Bucarest en el grafo definido en la Figura 12.



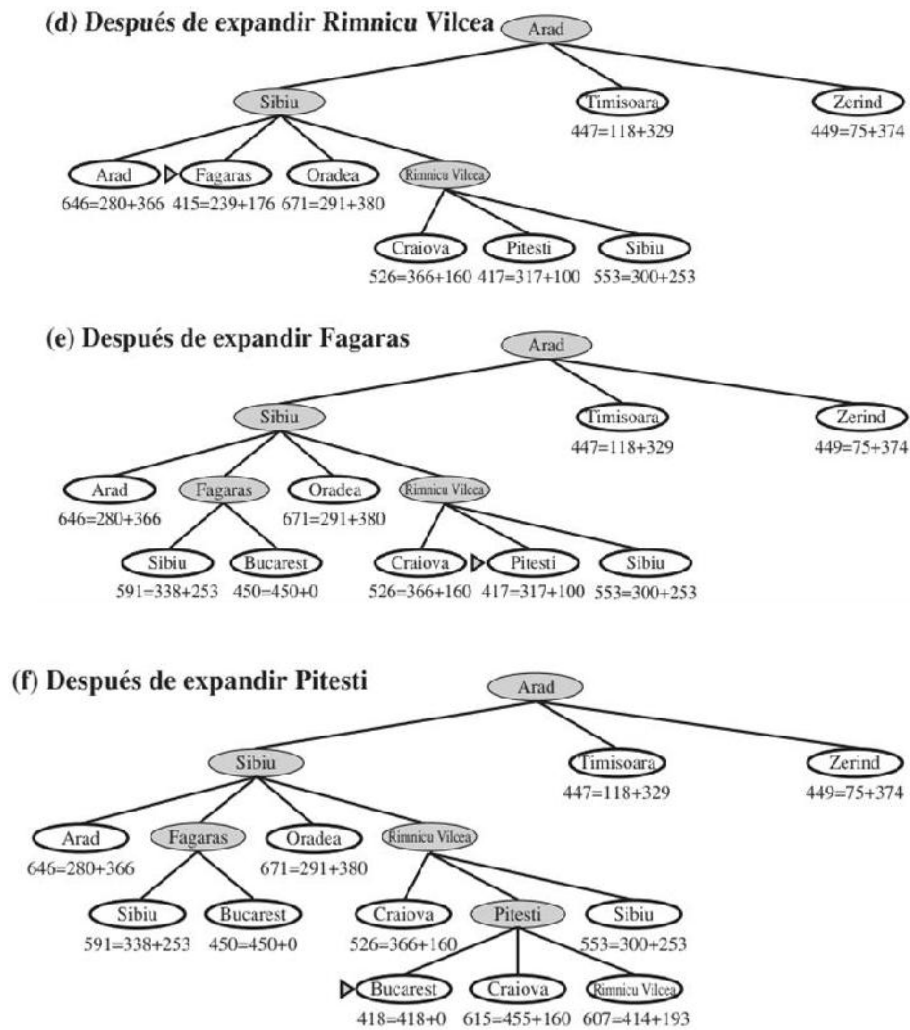


Figura 12: Árbol de búsqueda A* para Bucarest
Fuente: (Stuart J. Russell, & Peter Norvig, 2008, Inteligencia Artificial un Enfoque Moderno)

Descripción Figura 12:

“Etiquetamos los nodos con $f = g + h$; los valores h son las distancias en línea recta a Bucarest (heurística)”.

2.6. Geolocalización

La geolocalización es un término relativamente moderno que apareció a la par de los avances tecnológicos en el campo de la localización y hace referencia a la forma en que se

sitúan objetos o personas en el espacio mediante unas coordenadas de latitud, longitud y altura, reflejándose en un mapa. Su evolución ha sido posible gracias al desarrollo del Sistema de Posicionamiento Global (GPS) y su aplicabilidad gracias al desarrollo de tecnologías móviles, inalámbricas e Internet. (García, Fernández, Galvez, Sanchez, & Grimaldi, 2017, p.265)

2.7. Sistema de Posicionamiento Global

El Sistema de Posicionamiento Global o GPS, es un sistema global de navegación por satélite que permite determinar en todo el mundo la posición de un objeto, una persona, un vehículo o una nave. Podemos alcanzar una precisión hasta de centímetros, usando el GPS diferencial, pero lo habitual son unos pocos metros.

El GPS funciona mediante una red de 27 satélites (24 operativos y 3 de respaldo) en órbita a 20.200 km sobre el globo terráqueo, con trayectorias sincronizadas para cubrir toda la superficie de la Tierra. Cuando se desea determinar una posición, el receptor que se utiliza para ello localiza automáticamente como mínimo tres satélites de la red, de los que recibe unas señales indicando la posición y el reloj de cada uno de ellos. Con base en estas señales, el aparato sincroniza el reloj del Sistema de Posicionamiento y calcula el retraso de las señales; es decir, la distancia al satélite. Por "triangulación" los tres satélites calculan la posición en que el GPS se encuentra. La triangulación en el caso del Sistema de Posicionamiento Global se basa en determinar la distancia de cada satélite respecto al punto de medición. Conocidas las distancias, se determina fácilmente la propia posición relativa respecto a los tres satélites. Conociendo además las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtiene la posición absoluta o coordenadas reales del punto de medición. (Ros & Giménez, 2010, p.2)

2.8. Antecedentes

Federico Martínez E. y Gonzalo Sainz (2010) en la tesis realizada “MODELOS Y ALGORITMOS DE OPTIMIZACIÓN COMBINATORIA PARA PLANIFICACIÓN DE RUTAS EN REGATAS DE BARCOS DE VELA” en la Universidad de Buenos Aires, Argentina-Buenos Aires.

Se encontraron las siguientes conclusiones:

- La cota de VMC (Velocity Made Good to Course) descripta para el algoritmo A* exhibe un comportamiento muy interesante, explorando únicamente el área del grafo que es realmente pertinente a la navegación que se espera del barco. Este resultado, además de habernos permitido realizar experimentos de forma muy rápida para el ajuste de las heurísticas, puede ser transferido a las aplicaciones de ruteo de larga distancia convencional, a las que puede significarles una mejora significativa de rendimiento.
- En este trabajo implementamos un modelo más general que los utilizados en el software comercial, obteniendo así la posibilidad de extender su uso para situaciones de regatas de corta distancia en que el costo de las maniobras pueda resultar significativo.

Arias Rojas J. (2010) en la tesis realizada “APLICACIÓN DE UN MODELO DE OPTIMIZACIÓN EN LA PLANEACIÓN DE RUTAS DE LOS BUSES ESCOLARES DEL COLEGIO LICEO DE CERVANTES NORTE” *en la* Pontificia Universidad Javeriana, Bogotá-Colombia.

Se encontraron las siguientes conclusiones:

- Se exploró el proceso de transporte escolar del colegio Liceo de Cervantes y se analizaron sus características comparándolas con las variables y restricciones del problema de planeación de rutas para vehículos con capacidad. Esto permitió realizar una representación simplificada del proceso y modelarlo matemáticamente.
- El análisis financiero permitió visualizar la magnitud del costo del proceso de transporte y proyectar el ahorro que implicaría la reducción de la ruta.

Li Fernández M. y Fonseca Romero S. (2013) en la tesis realizada “DESARROLLO DE UNA HERRAMIENTA SOFTWARE PARA DETERMINAR LA RUTA DE ACARREO DE MINERAL MAS CORTA” en la Universidad Nacional de Trujillo, Trujillo-Perú.

Se encontraron las siguientes conclusiones:

- Luego de culminar el presente trabajo de investigación, concluimos que es posible desarrollar una herramienta de software capaz de determinar la ruta de acarreo de mineral más corta haciendo uso del método de conversión de rutas de acarreo Distancia Plana Equivalente.
- La Herramienta Software para determinar la Ruta de Acarreo de Mineral más Corta, comparado con el método tradicional, ofrece resultados efectivos en menor tiempo, lo que repercute en una reducción de costo para la empresa y les ayuda a sistematizar aún más sus procedimientos.

Meneses Pilco S. (2014) en la tesis realizada “DISEÑO DE UN ALGORITMO GENÉTICO PARA LA OPTIMIZACIÓN DE DISTANCIAS EN AMBIENTES TRIDIMENSIONALES” en la Pontificia Universidad Católica del Perú, Lima-Perú.

Se encontraron las siguientes conclusiones:

- Tanto el algoritmo Grasp como el algoritmo genético efectivamente resuelven el problema planteado inicialmente, permitiendo obtener una ruta a un bajo costo a partir de una instancia dada.
- La ruta que se obtiene a través de la ejecución del algoritmo Genético resulta mejor que aquella obtenida por el algoritmo Grasp.

Marchena Pejerrey D. (2015) en la tesis realizada “DESARROLLO DE UN SISTEMA PARA LA OPTIMIZACIÓN DE RUTAS DE TRABAJO UTILIZANDO EL ALGORITMO DE DIJKSTRA Y DIAGRAMAS DE VORONOI” en la Universidad Señor de Sipán, Pimentel-Perú.

Se encontraron las siguientes conclusiones:

- Se recopiló y ordenó los datos de los trabajos realizados por los operarios para poder aplicar los algoritmos de optimización de rutas y sectorización de zonas.
- Se logró aplicar el algoritmo de Dijkstra sobre los predios que conforman los puntos de acción, con el cual se hallaron las rutas óptimas requeridas para los itinerarios de trabajo; no obstante, se aplicó el algoritmo de Voronoi sobre el mapa catastral el cual realizó la sectorización de zonas de trabajo para que los operarios de campo tengan en cuenta el límite de hasta donde llegar a realizar su trabajo geográficamente.

Reyes Esquén J. (2018) en la tesis realizada “DESARROLLO DE UN PLANIFICADOR DE RUTAS PARA RECOJO DE DESECHOS SÓLIDOS UTILIZANDO ALGORITMO DE BELLMAN FORD” en la Universidad Señor de Sipán, Pimentel-Perú.

Se encontraron las siguientes conclusiones:

- El algoritmo Bellman-Ford logró generar un planificador de rutas para el recojo de desechos sólidos.

- Se generaron conjuntos de datos de geolocalización de la ciudad de Chiclayo que pueden ser utilizados por los vehículos recolectores de desechos sólidos.

2.9. Programas

2.9.1. Unity 3D

UNITY es un motor de desarrollo para la creación de contenidos 3D interactivos, con las características que es completamente integrado y que ofrece innumerables funcionalidades para facilitar el desarrollo de simulación y videojuegos.

El editor de Unity 3D se divide en 5 vistas principales:

- ***Explorador:*** Lista todos los elementos de tus proyectos. Permite ordenar de forma sencilla tu aplicación. En esta vista se encuentran las imágenes, escenas, scripts, audios, prefabs, texturas, atlas y todos los elementos que usarás o podrás usar en tu aplicación.
- ***Inspector:*** Muestra y define las propiedades de los elementos de tu proyecto. Modifica valores de forma rápida, cambia texturas, añade scripts, guarda prefabs, entre otros.
- ***Jerarquía:*** Lista jerárquica de los elementos de tu escena.
- ***Escena:*** Diseño y maqueta de tu contenido 3D completo o una pantalla o sección de este. Cada escena representa un nivel o sección diferente de la aplicación.
- ***Juego:*** Visualiza tu contenido 3D a distintas resoluciones. (Luttecke, 2014)

2.9.2. Blender

Blender es un software multiplataforma con versiones para los sistemas operativos Windows, Linux, MacOSX y FreeBSD; destinado al modelado 3D de objetos para después hacer representaciones de ese modelado.

Incorpora la posibilidad de dar texturas, materiales e iluminar la escena. Incluye las tecnologías más utilizadas en el diseño 3D: mallas, textos, meta objetos, curvas, superficies y modelado escultórico.

Permite crear animaciones de los modelados; desde mecánicas restringidas de artilugios robóticos hasta emisiones de partículas, pasando por todo tipo de cinemáticas para mover objetos o personajes. (Trammell, 2012)

2.9.3. Gimp

Gimp es el acrónimo de GNU Image Manipulation Program. Se trata de una aplicación de creación, composición y retoque de imágenes. Este programa de edición gráfica es el más conocido en las plataformas GNU/Linux, pero existen muchas otras plataformas sobre las que funciona, como: Microsoft, OpenBSD, FreeBSD, Solaris, SunOS, AIX, Digital Unix, OSF/1, IRIX, etc.

Mediante GIMP podemos crear y editar imágenes de mapas de bits principalmente. También podemos trabajar con imágenes vectoriales con el plug-in correspondiente, aunque no es el programa más adecuado para tratar imágenes vectoriales complejas. También podemos crear pequeñas animaciones mediante el tratamiento individual de las imágenes o tramas. GIMP también soporta la creación de pequeñas animaciones AVI y GIF. (GNU Image Manipulation Program [GIMP], 2005, p.3)

2.9.4. AutoCAD

AutoCAD es un programa de dibujo por computadora CAD 2 y 3 dimensiones, se puede crear dibujos o planos genéricos, documentar proyectos de ingeniería, arquitectura, mapas o sistemas de información geográfica por mencionar algunas industrias y aplicaciones. AutoCAD genera sus archivos en su propio formato DWG propietario de Autodesk, este es el programa pionero representante de la tecnología CAD (Computer Aided Design). (3D CAD Portal, 2018)

2.9.5. Xampp

XAMPP es un paquete de instalación independiente de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MariaDB, PHP, Perl.

El programa se distribuye bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y Mac OS X. (Sánchez & Parra, 2017, p.4)

2.9.6. Sublime Text

Sublime Text es un editor de texto con todas las funciones para editar archivos locales o una base de código. Eso incluye varias características para editar la base de código que ayuda a los desarrolladores a realizar un seguimiento de cambios. Varias características que son compatibles con Sublime son las siguientes:

- Resaltado de sintaxis

- Auto sangría
- Reconocimiento de tipo de archivo
- Barra lateral con los archivos del directorio mencionado.
- Macros
- Plug-in y paquetes

El editor de texto sublime se usa como un editor de desarrollo integrado (IDE) como Visual Studio Código y NetBeans. La versión actual del editor de texto sublime es 3.0 y es compatible con varios sistemas operativos como Windows, Linux y MacOS. (Tutorials Point, 2015, p.1)

2.10. Lenguajes de programación

2.10.1. C Sharp

C# es un lenguaje elegante, con seguridad de tipos y orientado a objetos, que permite a los desarrolladores crear una gran variedad de aplicaciones seguras y sólidas que se ejecutan en .NET Framework .NET. Puede usar C# para crear aplicaciones cliente de Windows, servicios web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos y muchas, muchas más cosas.

La sintaxis de C# es muy expresiva, pero también sencilla y fácil de aprender. Cualquier persona familiarizada con C, C++ o Java, reconocerá al instante la sintaxis de llaves de C#. Los desarrolladores que conocen cualquiera de estos lenguajes puede empezar normalmente a trabajar en C# de forma productiva en un espacio muy corto de tiempo. (Microsoft, 2015)

2.10.2. PHP

PHP, acrónimo de "Hypertext Preprocessor", es un lenguaje de 'scripting' de propósito general y de código abierto que está especialmente pensado para el desarrollo web y que puede ser embebido en páginas HTML. Su sintaxis recurre a C, Java y Perl, siendo así sencillo de aprender. El objetivo principal de este lenguaje es permitir a los desarrolladores web, escribir dinámica y rápidamente páginas web; aunque se puede hacer mucho más con PHP. (The PHP Group, 2019)

2.11. Otros lenguajes

2.11.1. HTML

HTML (HyperText Markup Language) es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia. La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, citas, etc) así como los diferentes efectos que se quieren dar y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado (como Mosaic, o Netscape). (Martínez, 2002, p.2)

2.11.2. CSS

El CSS (Cascading Style Sheets) es un lenguaje de estilos empleado para definir la presentación, el formato y la apariencia de un documento de marcaje, sea html, xml, o cualquier otro. Comúnmente se emplea para dar formato visual a documentos html o xhtml que funcionan como espacios web. También puede ser empleado en

formatos xml, u otros tipos de documentos de marcaje para la posterior generación de documentos.

Las hojas de estilos nacen de la necesidad de diseñar la información de tal manera que podemos separar el contenido de la presentación y, así, por una misma fuente de información, generalmente definida mediante un lenguaje de marcaje, ofrecer diferentes presentaciones en función de dispositivos, servicios, contextos o aplicativos. (Collell, 2015, p.7)

2.11.3. JavaScript

JavaScript es un lenguaje de scripts compacto basado en objetos (y no orientado a objetos). JavaScript permite la realización de aplicaciones de propósito general a través de la WWW y aunque no está diseñado para el desarrollo de grandes aplicaciones, es suficiente para la implementación de aplicaciones WWW completas o interfaces WWW hacia otras más complejas.

Por ejemplo, una aplicación escrita en JavaScript puede ser utilizada en un documento HTML proporcionando un mecanismo para el tratamiento de eventos, como un clic del ratón o bien la validación de entrada de datos en un formulario. Sin que exista una comunicación a través de la red una página HTML con JavaScript incrustado puede interpretar, y alertar al usuario con una ventana de diálogo, de que las entradas de los formularios no son válidas. O bien realizar algún tipo de acción como ejecutar un fichero de sonido o un applet de Java. (Universidad Nacional de San Luis, 2018, p.1)

2.12. Base de Datos

2.12.1. MySQL

MySQL es un sistema de administración de bases de datos relational (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización. MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger y hacer volcados de datos. Puede desarrollar sus propias aplicaciones de base de datos en la mayor parte de los lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos. MySQL utiliza el lenguaje de consulta estructurado (SQL). Se trata del lenguaje utilizado por todas las bases de relacionales, que presentaremos en una sección posterior. Este lenguaje permite crear bases de datos, así como agregar, manipular y recuperar datos en función de criterios específicos. (Gilfillan, 2003, p.40)

2.13. Proceso Unificado Ágil

El Proceso Unificado Ágil, es una metodología de desarrollo de software ágil heredada de otros paradigmas tales como la Programación Extrema (XP) y RUP (Proceso Unificado Rational) basado en procesos ágiles, desarrollada de por Scott Ambler que describe un enfoque simple, fácil entender, del desarrollo de software de aplicación de negocios usando técnicas y conceptos ágiles.

AUP aplica técnicas ágiles incluyendo desarrollo orientado a pruebas, modelado ágil, gestión de cambios ágil y refactorización de bases de datos para mejorar la productividad.

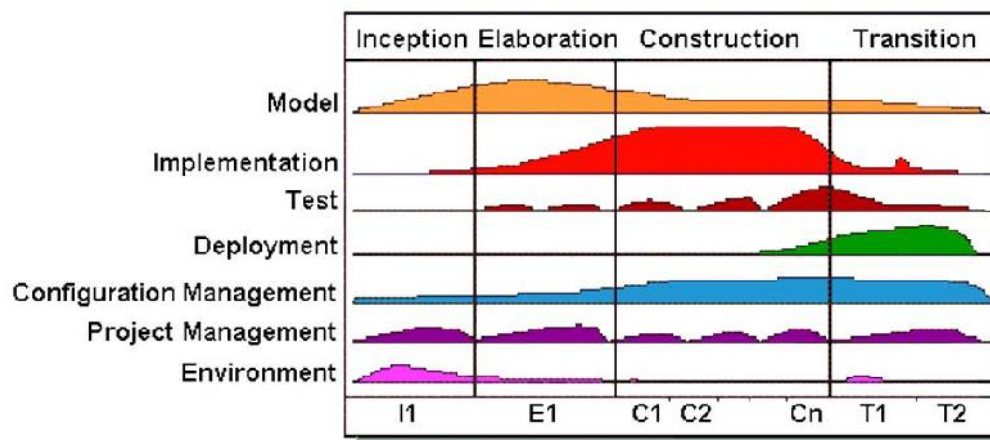


Figura 13. Ciclo de vida AUP
Fuente: (Núñez Mori José, 2010, Usabilidad en Metodologías Ágiles)

A continuación se describe las fases, las iteraciones y los entregables de la metodología AUP aplicada al siguiente proyecto:

INICIACIÓN:

El objetivo en esta fase es asimilar los requerimientos esperados de la solución y plasmarlos en la definición y especificación de los casos de uso. Asimismo, como apoyo a los procesos de gestión, se presenta la programación definitiva de las actividades y tareas conforme a la planificación del proyecto (diagrama de Gantt y WBS) junto con la relación de riesgos identificados. Los documentos como el catálogo de requerimientos, las especificaciones de requisitos de software, el cronograma del proyecto, la lista de riesgos, el plan de proyecto y enunciado de alcance se encuentran en observación durante esta fase.

- **Documento de Alcance del Proyecto.** Incluye definir en alto nivel lo que el sistema hará. Es también importante definir lo que el sistema no hará. Esto establece los límites en los que el equipo trabajará. Esto normalmente se representa mediante una lista de características o casos de uso.

- **Documento de Estimación de Costos y Planificación.** Se estima en alto nivel el calendario y el coste para el proyecto. Las estimaciones generales son usadas en iteraciones en las últimas fases, más específicamente se usan en las primeras iteraciones de la fase de elaboración.
- **Documento de Riesgos.** Los riesgos del proyecto se definen aquí. La gestión de riesgos es una parte importante de un proyecto AUP. La lista de riesgos cambia con el tiempo, a medida que se identifican nuevos riesgos, se mitigan, se evitan y se materializan y hay que ocuparse de ellos. Los riesgos de alta prioridad se tratan en fases más tempranas que los de baja prioridad.
- **Documento de Viabilidad.** El proyecto debe tener sentido desde las perspectivas técnicas, operacionales y de negocio. En otras palabras, deberíamos ser capaces de construirlo, una vez es desplegado deberíamos ser capaces de ponerlo en marcha, y debería tener un sentido económico el hacer estas cosas. Si el proyecto no es viable, debería ser cancelado.
- **Entorno del Proyecto.** Se tendrá que establecer que requerimientos de hardware y software son necesarios en corto y largo plazo para desarrollar la aplicación.

ELABORACIÓN:

En esta fase el objetivo es construir y probar la arquitectura descrita en el documento de arquitectura del sistema. Entre los entregables requeridos durante esta fase conviene citar

el documento de análisis (junto con el diagrama de clases de análisis) y el documento de diseño (acompañado del diagrama de clases de diseño):

- Seleccionar el uso de la tecnología, herramientas y recursos.
- Plantear el prototipo de arquitectura.
- Calcular el trabajo en corto tiempo.

CONSTRUCCIÓN:

En esta fase el objetivo es construir el software trabajando en forma iterativa e incremental, de acuerdo a los requerimientos del usuario. En esta fase se asignan las tareas de programación y la organización de la entrega.

En esta fase se debe realizar los siguientes hitos:

- Analizar y diseñar el modelo.
- Documentar las decisiones críticas de diseño.
- Construirlo.
- Ir evolucionando el dominio lógico, las interfaces de usuario y el esquema de datos.
- Probar el software.
- Desarrollar scripts para su instalación.
- Desarrollar una documentación inicial.
- Poner todos los productos bajo el CM (mantenimiento de configuración).

TRANSICIÓN:

Esta fase tiene como propósito la puesta del sistema en producción (afinando las pruebas integrales) junto a la capacitación de los usuarios y conversiones de sistemas en caso

existieran. A su vez se completará la documentación final del sistema. Los hitos a realizar son los siguientes:

- Finalizar la documentación.
- Detectar fallos.
- Validar el sistema.
- Validar la documentación.
- Finalizar el modelo de prueba.
- Finalizar la documentación.
- Finalizar el desarrollo del paquete.
- Colocar el sistema en producción.
- Formar a personas.

AUP tiene siete disciplinas:

Modelado. Entender el negocio de la organización, tratar el dominio del problema e identificar una solución viable para tratar el dominio del problema.

Implementación. Transformar el modelo en código ejecutable y realizar un nivel básico de pruebas, en particular pruebas unitarias.

Pruebas. Realizar una evaluación objetiva para asegurar calidad. Esto incluye encontrar defectos, validar que el sistema funciona como fue diseñado y verificar que se cumplen los requisitos.

Despliegue. Planificar el despliegue del sistema y ejecutar el plan para poner el sistema a disposición de los usuarios finales

Administración de la Configuración. Gestión de acceso a los artefactos del proyecto. Esto no sólo incluye el seguimiento de las versiones de los artefactos sino también controlar y gestionar los cambios en ellos.

Administración del Proyecto. Dirección de las actividades que tienen lugar dentro del proyecto. Esto incluye gestionar riesgos, dirigir a las personas y coordinar las personas y sistemas fuera del alcance del proyecto para asegurar que se entrega a tiempo y dentro del presupuesto.

Entorno. Soporte del resto del esfuerzo asegurando que el proceso, la orientación (estándares y guías) y las herramientas (software, hardware...) adecuadas están disponibles para el equipo cuando son necesarias.

AUP se basa en las siguientes filosofías:

1. Los empleados saben lo que están haciendo. La gente no va a leer documentación del proceso detallada, pero quieren algo de orientación a alto nivel y/o formación de vez en cuando. El producto AUP proporciona enlaces a muchos de los detalles pero no fuerza a ellos.
2. Simplicidad. Todo está descrito de forma concisa.
3. Agilidad. AUP se ajusta a los valores y principios de desarrollo de software ágil y la Alianza Ágil.
4. Foco en las actividades de alto valor. El foco está en las actividades que realmente cuentan, no en todas las posibles cosas que pudieran pasar en un proyecto.
5. Independencia de herramientas. Se puede usar cualquier conjunto de herramientas. La recomendación es que se usen las herramientas que mejor se adapten al trabajo, que son con frecuencia herramientas simples.
6. Habrá que adaptar AUP para cumplir con las necesidades propias.

AUP introduce propuestas como la programación por pares (todos los desarrolladores conocen el código implementado por todos), la gestión de requerimientos por niveles de

prioridad (toda solicitud de cambio es analizada y/o ejecutada durante la construcción del software) independencia entre herramientas para la concepción del producto y el refactoring o la modificación del código del programa sin alterar su comportamiento original mejorando en su estructura, performance y diseño. Asimismo propone el desarrollo dirigido por pruebas (TDD) a partir de un concepto denominado unidad de prueba (sincronizando tanto la construcción como las pruebas en el prototipo) de carácter reutilizable.

Las ventajas que presenta la metodología AUP, son las siguientes:

- Versión simplificada del RUP empleando técnicas de desarrollo ágiles.
- Preparados especialmente para cambios durante el desarrollo del proyecto (adaptables).
- Orientados más a las personas que a los procesos.
- El cliente es parte del equipo de desarrollo.
- Los integrantes saben lo que hacen. (Núñez, 2010)

CAPÍTULO III: Métodos y Materiales

3.1. Diseño de contratación de Hipótesis

Es un diseño no experimental transeccional o transversal, ya que estamos recopilando datos en un lugar y un momento dado, en este caso tomaríamos el momento dado en el mes de noviembre del año 2018 en la Universidad Nacional Pedro Ruiz Gallo, con respecto a la cantidad de alumnos que posee.

Para ser más específicos viene a ser un diseño no experimental transeccional de tipo explicativo, debido a que estamos recolectando datos sobre las incidencias de los alumnos, en la localización de las principales oficinas de la Universidad Nacional Pedro Ruiz Gallo.

Formalización:

$M_1 \text{ =====> } X \text{ =====> } M_2$

Dónde:

M_1 : Momento antes de desarrollar un sistema de búsqueda heurístico

X: Desarrollo de un sistema de búsqueda heurístico

M_2 : Momento después de desarrollar un sistema de búsqueda heurístico

3.2. Población y Muestra

3.2.1. Población

Se tomara como población a los alumnos de la Universidad Nacional Pedro Ruiz Gallo matriculados en el periodo 2018-II:

Según la oficina de Asuntos Académicos: 13790 aprox.

3.2.2. Muestra

Se tomara como muestra a los alumnos de la escuela profesional de Matemáticas matriculados en el curso Programación y Computación (2018-II):

Según la oficina de Asuntos Académicos: 63 alumnos.

3.3. Técnica de recolección de datos

La técnica que se utilizará en la investigación será:

La encuesta: Esta técnica fue aplicada a los alumnos seleccionados en la muestra, para recolectar información acerca del conocimiento que tienen en la ubicación de las principales oficinas de la Universidad Nacional Pedro Ruiz Gallo.

Observación: Esta técnica fue utilizada para medir los tiempos de recorrido entre personas que no conocían el destino a llegar (Facultad FACFyM) y una persona que si conocía el destino, que permita establecer las diferencias con respecto al tiempo de demora.

3.4. Instrumento de recolección de datos

Se tomó como instrumento de recolección de datos al cuestionario, que nos permitió recolectar datos por medio de preguntas. También se utilizó la guía de observación para medir tiempos de demora en el recorrido de una ruta.

CAPÍTULO IV: Resultados

4.1. INICIACIÓN

4.1.1. Descripción del Sistema

El sistema de búsqueda heurística es un software que tiene como objetivo principal ayudar a sus usuarios a movilizarse en la Universidad Nacional Pedro Ruiz Gallo. Este sistema estará compuesto principalmente por una aplicación móvil que interactuara con una interfaz de programación de aplicaciones (API), una base de datos y aplicación Web.

Su funcionamiento será el siguiente:

- La base de datos: se encargara de almacenar los datos utilizados en el sistema, organizados en cuatro registros: nodo, arista, destino y usuario.
- La API: Interactuara como capa intermedia entre la base de datos y la aplicación móvil; se encargara de brindar los datos necesarios para la ejecución de la búsqueda en forma organizada y simple.
- La aplicación web: se encargara de administrar los datos utilizados en la aplicación móvil, se asegurara de que la información mostrada se correcta, segura y este actualizada.
- La aplicación móvil: se encargara de realizar el proceso más importante del sistema, que es buscar las rutas hacia las principales oficinas de la Universidad Nacional Pedro Ruiz Gallo desde la ubicación del usuario por medio de la tecnología GPS. Esta búsqueda se realizara a través del algoritmo A*, que evalúa nodo a nodo un grafo bidireccional superpuestos en las vías peatonales de la universidad.

4.1.2. Consideraciones sobre el sistema

4.1.2.1. Representación de la localización

Para implementar el servicio de geolocalización en la aplicación móvil se seguirá los siguientes pasos:

Primero tendremos que tomar puntos geográficos de referencias que nos permitan delimitar el área de interés en el que será implantado este servicio, en nuestro caso el campus de la Universidad Nacional Pedro Ruiz Gallo.

Seguidamente tendremos que llevar a cabo la captura de las coordenadas de longitud y la latitud por medio del chip GPS del dispositivo móvil.

Una vez obtenido las coordenadas podremos situar la ubicación del dispositivo en el mapa de la universidad por medio de un cálculo entre las coordenadas y los puntos de referencia.

Finalmente podremos dar paso a la búsqueda de las rutas vinculando la ubicación del usuario al nodo del grafo más cercano.

4.1.2.2. Modelación 3D

Para realizar el modelo 3D de nuestra área de interés seguiremos los siguientes pasos:

- Adquirir el plano digitalizado del área a modelar, en nuestro caso el campus de la Universidad Nacional Pedro Ruiz Gallo en formato .dxf (Autocad).
- Recolectar fotografías de las estructuras de la universidad en todos los ángulos posibles.
- Exportar el plano digitalizado de la universidad al entorno de diseño 3D (blender) para ser usado como base de las estructuras tridimensionales.

- Diseñar las estructuras 3D proporcionándolas de volumen, forma y relieve.
- Finalmente texturizar los diseños con recortes de las fotografías recolectadas en el primer paso.

4.1.3. Alcance del Proyecto

4.1.3.1. Requerimientos

La captura y el análisis de los requerimientos del sistema es una de las fases más importantes para que un proyecto tenga éxito. “Como regla de modo empírico, el costo de reparar un error se incrementa en un factor de diez de una fase de desarrollo a la siguiente, por lo tanto la preparación de una especificación adecuada de requerimientos reduce los costos y el riesgo general asociado con el desarrollo” (Gómez, 2011, p.33).

A continuación se muestra los requerimientos identificados para el desarrollo del proyecto, según el Estándar 830 del Institute of Electrical and Electronics Engineers, (IEEE Std 830, 1998) “los requerimientos se dividen en dos grupos, funcionales y no funcionales. Los funcionales, que describirán las funciones que el software va a ejecutar; y los no funcionales, que especificarán criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus funciones específicas” (p.10).

Requerimientos Funcionales

Los requerimientos funcionales considerados para este proyecto son:

Tabla 5.

Requerimientos Funcionales, Módulo de Administración

Referencia	Requerimientos
RF1.1	El sistema debe pedir el acceso a la aplicación web por medio de una autenticación.
RF1.2	El sistema debe permitir crea, modificar e inactivar a los usuarios encargados de la administración.
RF1.3	El sistema debe permitir crear, modificar e inactivar los nodos que conforman el grafo.
RF1.4	El sistema debe permitir crear, modificar e inactivar las aristas que conectan los nodos.
RF1.5	El sistema debe permitir crea, modificar e inactivar los destinos que serán objeto de búsqueda en la aplicación móvil.
RF1.6	El sistema debe permitir visualizar el diseño del grafo integrado en el mapa de la Universidad Nacional Pedro Ruiz Gallo.
RF1.7	El sistema debe permitir mostrar la cantidad de usuarios que utilizan la aplicación móvil.

Fuente: Elaboración propia

Tabla 6.

Requerimientos Funcionales, Módulo de Usuario

Referencia	Requerimientos
RF1.8	El sistema debe importar los datos requeridos para la búsqueda de rutas.
RF1.9	El sistema debe permitir seleccionar el origen de la búsqueda.
RF1.10	El sistema debe permitir seleccionar el destino a buscar.
RF1.11	El sistema debe buscar una ruta entre la ubicación del usuario y el destino seleccionado, y trazarlo en el mapa de la universidad Nacional Pedro Ruiz Gallo.

Fuente: Elaboración propia

Requerimientos No Funcionales

Los requerimientos no funcionales considerados para este proyecto son:

Tabla 7.

Requerimientos no Funcionales

Referencia	Requerimientos
RNF2.1	El sistema deberá estar disponible para ser usado en la plataforma Android por los usuarios y Web por la administración.
RNF2.2	El sistema será desarrollado con las herramientas Unity 3D, Sublime Tex y Xampp.
RNF2.3	La aplicación Web será almacenada en un Web Hosting Apache.
RNF2.4	El sistema considerará una arquitectura cliente - servidor.
RNF2.5	El sistema tendrá un base de datos relacional: los datos se organizan en tablas compuestas por filas y columnas
RNF2.6	Las contraseñas de los usuarios deberán ser encriptados con MD5.
RNF2.7	El sistema será codificado con los lenguaje de programación C# y PHP.
RNF2.8	El sistema tendrá una API que comunicara a la aplicación móvil con la base de datos.

Fuente: Elaboración propia

4.1.3.2. Diagrama y Especificación de Casos de Uso

El proyecto está dividido en 11 casos de uso y se han identificado como actores del sistema al Usuario Administrador, Colaborador y Móvil:

Iniciar Sesión: El propósito de este caso de uso es el de permitir al usuario administrador y colaborador acceder a la página web encargada de administrar la aplicación móvil.

Gestionar Usuario: El propósito de este caso de uso es el registro de nuevos usuarios colaboradores que apoyen con la gestión de la aplicación móvil. Además, el usuario administrador es el responsable del mantenimiento de las cuentas de usuario.

Gestionar Nodo: El propósito de este caso de uso es la creación y mantenimiento de los nodos que conforman el grafo.

Gestionar Arista: El propósito de este caso de uso es el registro y mantenimiento de las aristas que vinculan a los nodos.

Gestionar Destino: El propósito de este caso de uso es el registro y mantenimiento de los destinos que serán objeto de búsqueda en el grafo.

Generar Grafo: El propósito de este caso de uso es mostrar el diseño del grafo superpuesto en el mapa de la Universidad Nacional Pedro Ruiz Gallo, para que facilite la gestión.

Generar Reporte Visitantes: El propósito de este caso de uso es mostrar la cantidad de usuarios móviles que utilizan la aplicación móvil.

Cargar Datos: El propósito de este caso de uso es importar los datos requerido para la realización de la búsqueda de rutas en la aplicación móvil.

Consultar Origen: El propósito de este caso de uso es la consulta los puntos de origen disponibles.

Consultar Destino: El propósito de este caso de uso es la consulta de los destinos disponibles.

Buscar Ruta: El propósito de este caso de uso es la generación de rutas desde la ubicación del usuario móvil y el destino seleccionado.

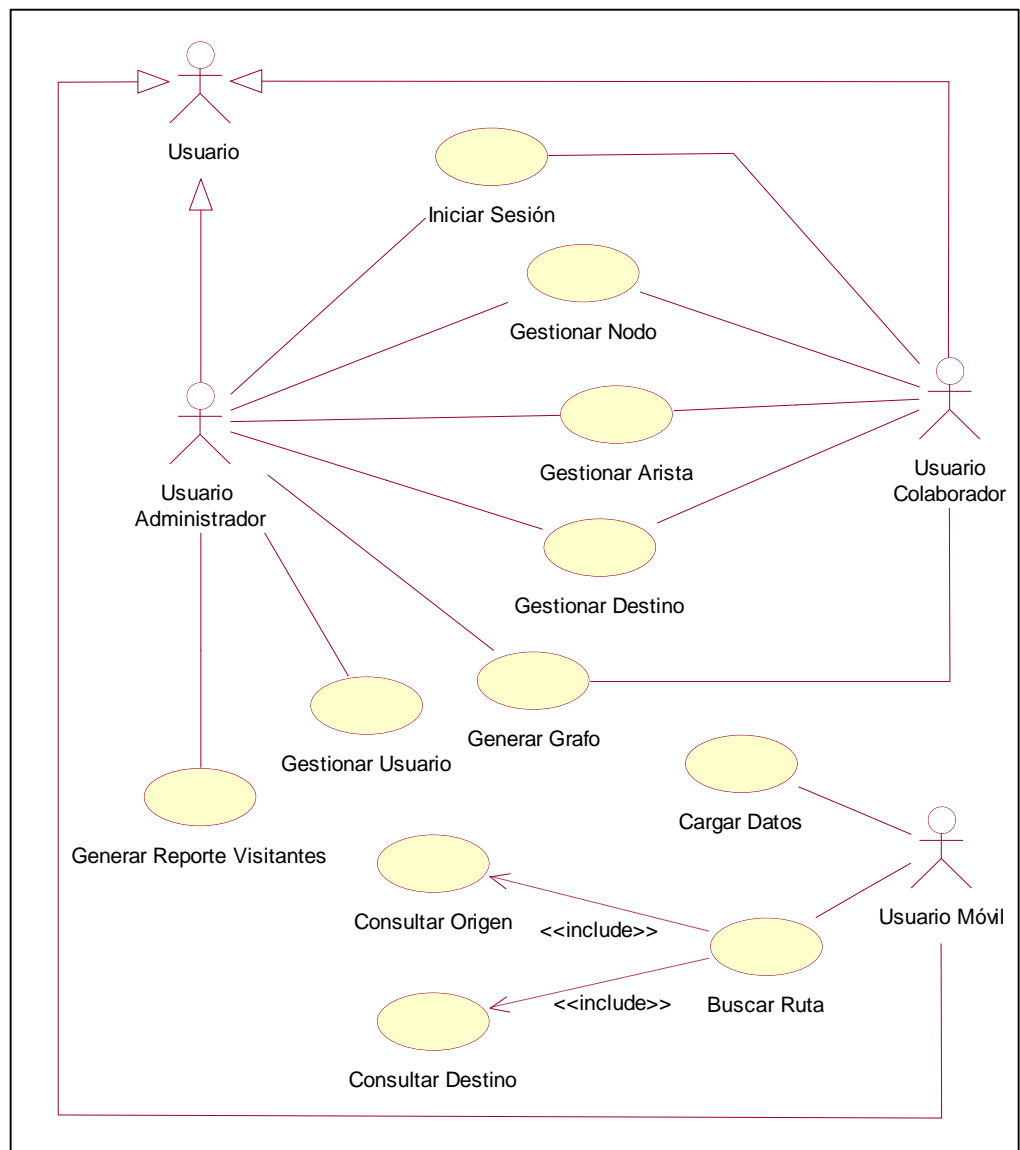


Figura 14. Diagrama de caso de uso del sistema

Fuente: Elaboración Propia

Especificación de los casos de uso

En esta parte, se realiza una descripción detallada de los casos de uso donde incluyen: precondiciones, pos condiciones, flujo de eventos.

Tabla 8.

Especificación de Caso de Uso, Iniciar Sesión

Nombre de Caso de Uso		Iniciar Sesión
Objetivo	:	Permitir el acceso a la página web
Actor Principal	:	Usuario Administrador, Colaborador
Precondiciones	:	Estar registrado en la base de datos
Flujo Principal	:	<ul style="list-style-type: none">▪ El usuario ingresa los datos de Id y contraseña.▪ El sistema verifica si el usuario se encuentra registrado.▪ En caso de estar registrado, el sistema permite visualizar al usuario los módulos correspondientes; caso contrario, se mostrará un mensaje de “usuario o contraseña incorrecta”.
Flujo Alternativo	:	Verificar conexión a internet
Post Condiciones	:	Si los datos ingresados son validados por el sistema y se ha seguido el flujo principal de eventos, se visualiza los módulos del sistema.

Fuente: Elaboración propia

Tabla 9.

Especificación de Caso de Uso, Gestionar Usuario

Nombre de Caso de Uso	Gestionar Usuario
Objetivo	: Registro y mantenimiento de los usuarios
Actor Principal	: Usuario Administrador
Precondiciones	: Iniciar sesión <ul style="list-style-type: none"> ▪ El usuario administrador debe acceder al módulo de principal y seleccionar “Gestionar Usuario”. ▪ Para crear un nuevo Usuario, el Administrador presionara el botón “Nuevo” para llenar los datos del Usuario a crear. ▪ El usuario Administrador presiona el botón “Guardar”.
Flujo Principal	: <ul style="list-style-type: none"> ▪ El sistema verifica en la base de datos, los datos ingresados por el Administrador. ▪ Si los datos ingresados aún no se encuentran registrados en la base de datos, el sistema crea el nuevo Usuario, de lo contrario se muestra un mensaje de “Usuario Existente”. ▪ Para un usuario ya existente, el administrador podrá pulsar el botón “Modificar” o “Inhabilitar” que le corresponda, para poder editar los datos del usuario o inhabilitarla.
Flujo Alternativo	: Verificar conexión a internet Si los datos ingresados son validados por el sistema y se ha seguido el flujo principal de
Post Condiciones	: eventos, el formulario queda habilitado para el registro de una nuevo Usuario.

Fuente: Elaboración propia

Tabla 10.

Especificación de Caso de Uso, Gestionar Nodo

Nombre de Caso de Uso	Gestionar Nodo
Objetivo	: Registro y mantenimiento de los nodos
Actor Principal	: Usuario Administrador, Colaborador
Precondiciones	: Inicia Sesión <ul style="list-style-type: none"> ▪ El usuario debe acceder al módulo de principal y seleccionar “Gestionar Nodo”. ▪ Para crear un nuevo nodo, el usuario presionara el botón “Nuevo” para llenar los datos del nodo a crear. ▪ El usuario presiona el botón “Guardar”. ▪ El sistema verifica en la base de datos, los datos ingresados por el usuario.
Flujo Principal	: <ul style="list-style-type: none"> ▪ Si los datos ingresados aún no se encuentran registrados en la base de datos, el sistema crea el nuevo nodo, de lo contrario se muestra un mensaje de “Nodo Existente”. ▪ Para un nodo ya existente, el usuario podrá pulsar el botón “Modificar” o “Inhabilitar” que le corresponda, para poder editar los datos del nodo o inhabilitarlo.
Flujo Alternativo	: Verificar conexión a internet Si los datos ingresados son validados por el sistema y se ha seguido el flujo principal de
Post Condiciones	: eventos, el formulario queda habilitado para el registro de una nuevo Nodo.

Fuente: Elaboración propia

Tabla 11.

Especificación de Caso de Uso, Gestionar Destino

Nombre de Caso de Uso	Gestionar Destino
Objetivo	: Registro y mantenimiento de los Destinos
Actor Principal	: Usuario Administrador, Colaborador
Precondiciones	: Deben existir nodos registrados <ul style="list-style-type: none"> ▪ El usuario debe acceder al módulo de principal y seleccionar “Gestionar Destino”. ▪ Para crear un nuevo destino, el usuario presionara el botón “Nuevo” para llenar los datos del destino a crear. ▪ El usuario presiona el botón “Guardar”. ▪ El sistema verifica en la base de datos, los datos ingresados por el usuario.
Flujo Principal	: <ul style="list-style-type: none"> ▪ Si los datos ingresados aún no se encuentran registrados en la base de datos, el sistema crea el nuevo destino, de lo contrario se muestra un mensaje de “Destino Existente”. ▪ Para un destino ya existente, el usuario podrá pulsar el botón “Modificar” o “Inhabilitar” que le corresponda, para poder editar los datos del destino o inhabilitarlo.
Flujo Alternativo	: Verificar conexión a internet Si los datos ingresados son validados por el sistema y se ha seguido el flujo principal de eventos, el
Post Condiciones	: formulario queda habilitado para el registro de un nuevo Destino.

Fuente: Elaboración propia

Tabla 12.

Especificación de Caso de Uso, Gestionar Arista

Nombre de Caso de Uso	Gestionar Arista
Objetivo	: Registro y mantenimiento de las aristas
Actor Principal	: Usuario Administrador, Colaborador
Precondiciones	: Inicia Sesión <ul style="list-style-type: none"> ▪ El usuario debe acceder al módulo de principal y seleccionar “Gestionar Arista”. ▪ Para crear una nueva arista, el usuario presionara el botón “Nuevo” para llenar los datos de la arista a crear. ▪ El usuario presiona el botón “Guardar”. ▪ El sistema verifica en la base de datos, los datos ingresados por el usuario.
Flujo Principal	: <ul style="list-style-type: none"> ▪ Si los datos ingresados aún no se encuentran registrados en la base de datos, el sistema crea la nueva arista, de lo contrario se muestra un mensaje de “Arista Existente”. ▪ Para una arista ya existente, el usuario podrá pulsar el botón “Modificar” o “Inhabilitar” que le corresponda, para poder editar los datos de la arista o inhabilitarla.
Flujo Alternativo	: Verificar conexión a internet Si los datos ingresados son validados por el sistema y se ha seguido el flujo principal de
Post Condiciones	: eventos, el formulario queda habilitado para el registro de una nueva arista.

Fuente: Elaboración propia

Tabla 13.

Especificación de caso de uso, Generar Grafo

Nombre de Caso de Uso		Generar Grafo
Objetivo	:	Visualizar el diseño del grafo
Actor Principal	:	Usuario Administrador, Colaborador
Precondiciones	:	Inicia Sesión <ul style="list-style-type: none"> El usuario debe acceder al módulo principal y
Flujo Principal	:	seleccionar “Generar Grafo” para visualizar el diseño del grafo.
Flujo Alternativo	:	Verificar conexión a internet
Post Condiciones	:	Ninguna

Fuente: Elaboración propia

Tabla 14.

Especificación de Caso de Uso, Cargar Datos

Nombre de Caso de Uso		Cargar Datos
Objetivo	:	Importar los datos requeridos para la búsqueda de rutas en la aplicación móvil.
Actor Principal	:	Usuario móvil
Precondiciones	:	Ninguna <ul style="list-style-type: none"> El usuario móvil pulsa el botón “Iniciar” para
Flujo Principal	:	importar los datos requeridos para la búsqueda de rutas: nodos, aristas y destinos.
Flujo Alternativo	:	Verificar conexión a internet
Post Condiciones	:	Ninguna

Fuente: Elaboración propia

Tabla 15.

Especificación de Caso de Uso, Buscar Ruta

Nombre de Caso de Uso		Buscar Ruta
Objetivo	:	Trazar una ruta desde la ubicación del usuario al destino seleccionado.
Actor Principal	:	Usuario móvil
Precondiciones	:	Ninguna
Flujo Principal	:	<ul style="list-style-type: none"> ▪ El usuario móvil debe acceder al módulo “Buscar Ruta”. ▪ El usuario móvil debe pulsar el botón “Ubicación” para obtener las coordenadas de su ubicación. ▪ El usuario móvil debe pulsar el botón “Destinos” para ubicar los destinos existentes. ▪ El usuario móvil debe pulsar el botón correspondiente al destino deseado para cargar sus datos. ▪ El usuario móvil debe pulsar el botón “Buscar” para trazar una ruta desde la ubicación del dispositivo al destino seleccionado.
Flujo Alternativo	:	Verificar conexión a internet
Post Condiciones	:	Si se ha seguido el flujo principal de eventos, el formulario queda habilitado para volver a generar una nueva Ruta.

Fuente: Elaboración propia

Tabla 16.

Especificación de Caso de Uso, Consultar Destino

Nombre de Caso de Uso		Consultar Destino
Objetivo	:	Seleccionar el destino a localizar.
Actor Principal	:	Usuario móvil
Precondiciones	:	Deben existir destinos registrados <ul style="list-style-type: none"> ▪ El usuario móvil debe acceder al módulo “Buscar Ruta”. ▪ El usuario móvil debe pulsar el botón
Flujo Principal	:	“Destinos” para ubicar los destinos existentes. <ul style="list-style-type: none"> ▪ El usuario móvil debe pulsar el botón correspondiente al destino deseado para cargar sus datos.
Flujo Alternativo	:	Verificar conexión a internet
Post Condiciones	:	Ninguna

Fuente: Elaboración propia

Tabla 17.

Especificación de Caso de Uso, Consultar Origen

Nombre de Caso de Uso		Consultar Origen
Objetivo	:	Seleccionar el punto de partida.
Actor Principal	:	Usuario móvil
Precondiciones	:	Deben existir puntos de origen registrados <ul style="list-style-type: none"> ▪ El usuario móvil debe acceder al módulo “Buscar Ruta”. ▪ El usuario móvil debe pulsar el botón “Origen”
Flujo Principal	:	para ubicar los orígenes existentes. <ul style="list-style-type: none"> ▪ El usuario móvil debe pulsar el botón correspondiente al origen deseado para cargar sus datos.
Flujo Alternativo	:	Verificar conexión a internet
Post Condiciones	:	Ninguna

Fuente: Elaboración propia

Tabla 18.

Especificación de caso de uso, Generar Reporte Visitantes

Nombre de Caso de Uso	Generar Reporte Visitantes
Objetivo	: Visualizar la cantidad de usuarios que utilizan el aplicativo móvil.
Actor Principal	: Usuario Administrador
Precondiciones	: Inicia Sesión <ul style="list-style-type: none"> ▪ El usuario administrador debe acceder al módulo principal y seleccionar “Generar Reporte Visitantes”.
Flujo Principal	: <ul style="list-style-type: none"> ▪ El usuario administrador debe seleccionar una fecha y luego presionar el botón “Consultar” para mostrar el número de usuarios que utilizaron el aplicativo móvil.
Flujo Alternativo	: Verificar conexión a internet
Post Condiciones	: Ninguna

Fuente: Elaboración propia

4.1.3.3. Diagrama WBS (Work Breakdown Structure)

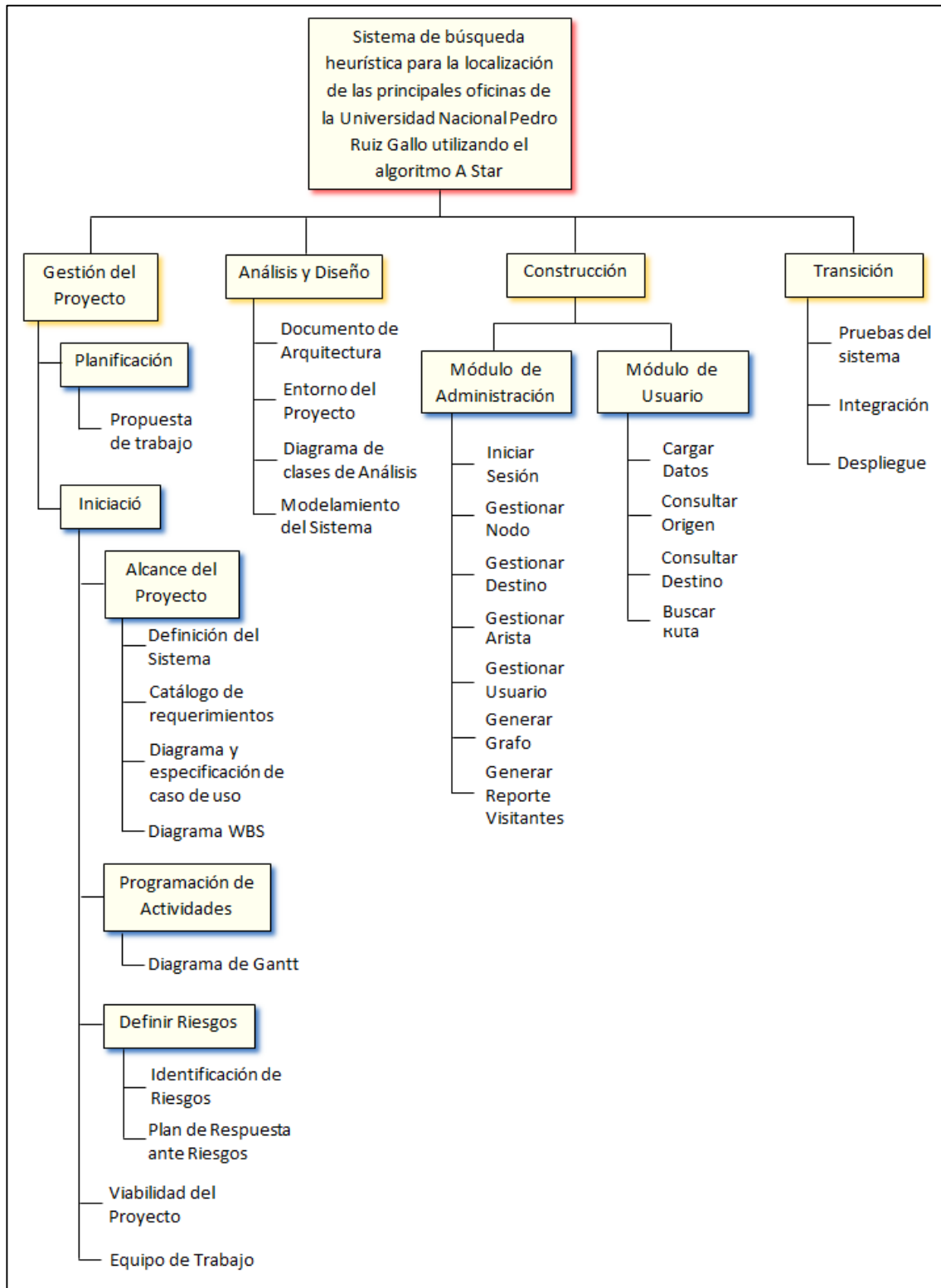


Figura 15. Diagrama WBS
Fuente: Elaboración Propia

4.1.4. Programación de Actividades

4.1.4.1. Diagrama de Gantt

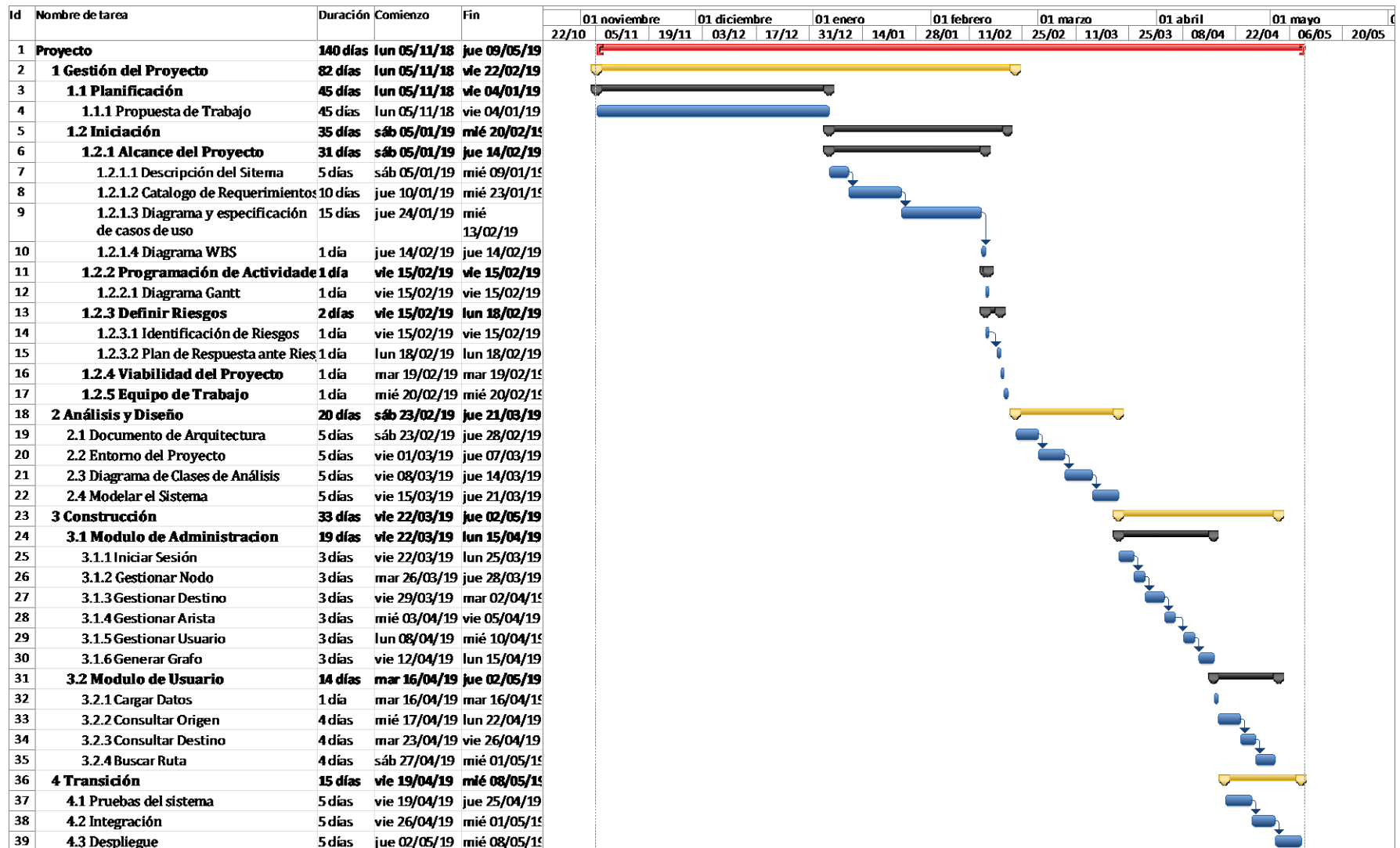


Figura 16. Diagrama de Gantt

Fuente: Elaboración Propia

4.1.5. Definición de Riesgos

4.1.5.1. Identificación de riesgos

De acuerdo con el Project Management Institute, (2013). Señala que: “El riesgo en un proyecto se define como un evento o condición incierta que, de producirse, tiene un efecto positivo o negativo en uno o más de los objetivos del proyecto, tales como el alcance, el cronograma, el costo y la calidad” (p.310).

Para el desarrollo del sistema se determinaron en cada fase de la metodología AUP (Iniciación, Construcción, Elaboración y Transición) los siguientes riesgos potenciales que debieron tomarse en cuenta para llevar a cabo una gestión apropiada.

En la Tabla 20 se muestran los riesgos identificados y clasificados en la Matriz de Probabilidad e Impacto (MPI), permitiendo relacionar los eventos considerados como riesgos con el grado de probabilidad de ocurrencia e impacto respecto al proyecto en su conjunto. Finalmente, la última columna refleja el grado de importancia del riesgo.

Para la clasificación de cada dimensión de la Matriz de Probabilidad e Impacto se asumieron las escalas mostradas en la Tabla 19.

Tabla 19.
Matriz de Probabilidad e Impacto

	Probabilidad		Impacto		
	Muy leve	Leve	Moderado	Severo	Muy severo
Muy Alta	Riesgo bajo	Riesgo moderado	Riesgo alto	Riesgo alto	Riesgo alto
Alta	Riesgo bajo	Riesgo moderado	Riesgo moderado	Riesgo alto	Riesgo alto
Media	Riesgo bajo	Riesgo bajo	Riesgo moderado	Riesgo alto	Riesgo alto
Baja	Riesgo bajo	Riesgo bajo	Riesgo moderado	Riesgo moderado	Riesgo alto
Muy baja	Riesgo bajo	Riesgo bajo	Riesgo bajo	Riesgo bajo	Riesgo moderado

Fuente: Elaboración propia

Tabla 20.
Identificación de Riesgos

Fases AUP	Riesgo	Probabilidad	Impacto	Importancia
Iniciación	Requerimientos incompletos o ambiguos	Alta	Muy severo	Riesgo alto
	Ausencia del usuario en el levantamiento de los requerimientos	Media	Severo	Riesgo alto
	Cambio en los requerimientos durante el proyecto	Muy Alta	Severo	Riesgo alto

Fuente: Elaboración propia

Continuación de la Tabla 20, Identificación de Riesgos

Construcción	Falta de especificación de la arquitectura	Media	Muy severo	Riesgo alto
	Incorrecta definición y estructura de los datos establecidos	Muy Baja	Moderado	Riesgo bajo
Elaboración	Integración compleja de los módulos del software	Alta	Severo	Riesgo alto
	Modificación del cronograma de actividades	Muy Alta	Moderado	Riesgo alto
	Recursos de hardware con rendimiento insatisfactorio para la elaboración.	Media	Severo	Riesgo alto
	El software es complejo de implementar	Media	Moderado	Riesgo moderado
Transición	Alcance de las pruebas, no definidas completamente	Alta	Muy severo	Riesgo alto
	Realización incompleta de las pruebas	Media	Severo	Riesgo alto
	Presencia de errores en el software al instalarse en el ambiente de producción	Media	Muy severo	Riesgo alto

Fuente: Elaboración propia

4.1.5.2. Plan de respuesta ante riesgos

Existen varias estrategias de respuesta a los riesgos. Para cada riesgo, se debe seleccionar la estrategia o la combinación de estrategias con mayor probabilidad de eficacia.

En el presente proyecto se tomaron en cuenta principalmente “las cuatro estrategias que normalmente abordan las amenazas o los riesgos que pueden tener impactos negativos sobre los objetivos de un proyecto en caso de materializarse, son: evitar, transferir, mitigar y aceptar” (Project Management Institute, 2013, p.344). Estas estrategias son asignadas en función a la importancia de cada riesgo, definidos en la sección anterior. Las estrategias de evitar y mitigar serán aplicadas a los riesgos definidos como altos, mientras que las de transferir y aceptar serán aplicadas a riesgos con menor trascendencia.

Tabla 21.
Respuesta ante Riesgos

Riesgo	Estrategia	Respuesta
Requerimientos incompletos o ambiguos	Mitigar	Capacitación del equipo de desarrollo sobre lógica de negocio
Ausencia del usuario en el levantamiento de los requerimientos	Mitigar	Participación del usuario en la definición de requerimientos
Cambio en los requerimientos durante el proyecto	Evitar	Selección de una metodología ágil que permita aplacar los cambios en los requerimientos con mayor facilidad

Fuente: Elaboración propia

Continuación de la Tabla 21, Respuesta ante Riesgos

Falta de especificación de la arquitectura	Evitar	Utilizar modelos, vistas y diagramas para el diseño de la arquitectura
Incorrecta definición y estructura de los datos establecidos	Aceptar	Consultar material bibliográfico y fuentes confiables durante la etapa de construcción
Integración compleja de los módulos del software	Mitigar	Realizar programación en equipo y desarrollo modular paralelamente
Modificación del cronograma de actividades	Evitar	Realizar una buena planeación de recursos, tareas y tiempos para evitar posibles desfases
Recursos de hardware con rendimiento insatisfactorio para la elaboración.	Mitigar	Gestionar por parte del equipo de desarrollo con anterioridad los recursos de hardware necesarios para poder ejecutarse las tareas planificadas
El software es complejo de implementar	Transferir	Consultar la solución mediante opiniones de un experto
Alcance de las pruebas, no definidas completamente	Evitar	Realizar reuniones contextuales con el desarrollador y el analista
Realización incompleta de las pruebas	Evitar	Asignar a un tester diferente al desarrollador que utilice buenas prácticas y técnicas de pruebas unitarias, que logre evaluar totalmente el software en un periodo corto de tiempo
Presencia de errores en el software al instalarse en el ambiente de producción	Mitigar	Realizar pruebas piloto en el ambiente de producción

Fuente: Elaboración propia

4.1.6. Viabilidad del Proyecto

En este aspecto, se deben considerar los costos incurridos en el desarrollo de la aplicación, ya que se presentan restricciones tanto en hardware como en software:

Tabla 22.
Viabilidad del Proyecto

Requerimiento	Viabilidad Técnica	Viabilidad Económica
Disponibilidad de un equipo de cómputo para las labores de análisis, diseño, construcción y pruebas.	Se cuenta con una computadora portátil Intel Core i3 de 3.30 GHz con 4 GB de RAM ofreciendo así un rendimiento superior para las fases de análisis, diseño, desarrollo y pruebas.	Requerimiento fuera del costo ya que su aprovisionamiento es asumido bajo la responsabilidad de los tesistas.
Herramientas CASE de libre distribución para el modelamiento UML y construcción de la base de datos de la solución.	Se encuentra cubierto por el uso de la herramienta StarUML sujetos a las exigencias técnicas propias de la documentación AUP.	No tienen restricción de costo debido a que la herramienta a utilizar es de uso libre y sin licencia.
Herramienta IDE para la construcción de la interfaz gráfica y codificación de la aplicación móvil bajo la plataforma Android	Se encuentra cubierto por el uso del motor de desarrollo Unity 3D.	No tienen restricción de costo debido a que se utilizara la edición personal de Unity de uso libre y gratuito para estudiantes y desarrolladores independientes.
Herramienta Editor de código multiplataforma para la construcción de la página Web	Se encuentra cubierto por el uso del editor de texto sublime Text.	No tienen restricción de costo debido a que la herramienta a utilizar es de uso libre y sin licencia.

Fuente: Elaboración propia

Continuación de la Tabla 22, Viabilidad del Proyecto

Sistema administrador de base de datos de libre distribución con capacidad para soportar múltiples conexiones.	Es cubierto bajo la elección del servidor MySQL	No tienen restricción de costo debido a que la herramienta a utilizar es de uso libre y sin licencia.
Disponibilidad de un servidor Web para labores de implementación.	Es cubierto bajo la elección del servidor Apache	No tienen restricción de costo debido a que la herramienta a utilizar es de uso libre y sin licencia.
El lenguaje de programación para la construcción bajo el paradigma orientado a objetos	Es constituido bajo la elección de los lenguajes de programación C# y PHP	No tienen restricción de costo debido a que su disposición está incluida en el IDE.

Fuente: Elaboración propia

La Tabla 23 muestra el costo asumido por concepto del personal durante la realización del proyecto. Del mismo modo la Tabla 24 resume la inversión realizada en cada fase de proyecto con un horizonte de cinco meses, expresada en soles.

*Tabla 23.
Costo de Personal*

Rol	Cantidad	Costo /Hora (/S.)
Analista	1	5
Programador	1	5
Tester	1	5

Fuente: Elaboración propia

Tabla 24.
Costo referencia del Proyecto

Fase	Responsable	Horas estimadas	Costo (S/.)
Iniciación	Analista	656	3280
Elaboración	Analista	160	800
Construcción	Programador	264	1320
Transición	Tester	120	600
Total			6000

Fuente: Elaboración propia

Para el desarrollo del proyecto, no se produjeron inversiones de dinero respecto a pago de personal debido a que el desarrollo del proyecto es de completa responsabilidad de los tesistas.

Por lo tanto el proyecto se puede considerar técnica y económica viable.

4.1.7. Asignación de funciones a Hardware y software

Las funciones asignadas al hardware durante el desarrollo del proyecto son:

- Como computadora personal, albergar aplicaciones, IDE, CASE y ofimáticas requeridas para labores de análisis, diseño, construcción y pruebas.
- Como servidor, cumplir con el almacenamiento físico de la aplicación Web y la base de datos.

Las funciones asignadas al software durante el proyecto son:

- Asistir en las actividades de diagramación, modelamiento y documentación durante las fases de análisis y diseño.
- Permitir la codificación óptima y eficiente de los módulos, componentes y funcionalidades de la solución.

- Permitir la construcción de la interfaz gráfica de la aplicación Móvil por medio de la función “drag and drop”.
- Permitir la construcción de la interfaz gráfica de la aplicación Web vía código HTML.
- Proporcionar la simulación de los servidores web y de base de datos durante la fase de construcción.

4.2. ELABORACIÓN

4.2.1. Identificación de la Arquitectura

Para el desarrollo del presente proyecto se utilizará el modelo Cliente - Servidor. Esta arquitectura será una red de comunicaciones que conectará a distintos clientes a un servidor central y a un servidor de base de datos, es decir, una arquitectura de tres niveles. Los servidores cumplirán la función de proveer de recursos al cliente, mientras que el cliente solicitará dichos recursos por medio de peticiones al servidor central, quien otorgará una respuesta.

El modelo cliente – servidor propuesto se muestra a continuación en la Figura 17.

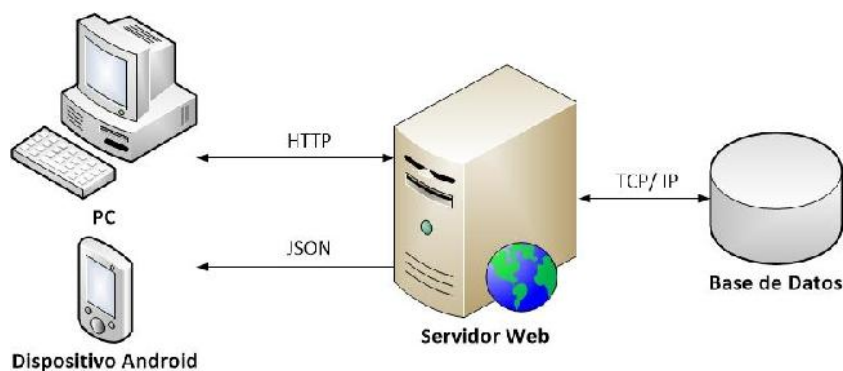


Figura 17. Arquitectura Cliente – Servidor
Fuente: Elaboración Propia

- **Dispositivo Android:**

Se encarga de mostrar la interfaz final al usuario, quien podrá navegar y ver información según los requerimientos establecidos.

- **PC:**

Computadora que permite acceder a la interfaz de administración.

- **Servidor Web:**

Sistema de administración y base de datos del sistema.

- **Servidor Base de datos:**

Servidor principal del sistema, se encarga de suministrar la información necesaria al dispositivo móvil, proveer al usuario administrador del sistema de administración web y comunicarse con la base de datos.

Concluyendo con la identificación de la arquitectura. El modelo cliente - servidor propuesto presenta 2 tipos de clientes:

- **Cliente liviano**

Este cliente esta representado por la aplicación, puesto que depende del servidor central para realizar tareas de procesamiento. Su objetivo principal es transportar la entrada y la salida entre el usuario y el servidor remoto.

- **Cliente Robusto**

Este cliente esta representado por la aplicación móvil instalada en el dispositivo android, ya que asume todas las funcionalidades y procesos, y solo deja al servidor la tarea de la gestión de transacciones con la base de datos.

4.2.2. Vista lógica de la Arquitectura

La vista lógica describe la estructura interna del sistema, es decir las partes que lo integran y como interactúan.

La vista lógica de la arquitectura propuesta esta representada en la figura 18 y muestra las capas lógicas y componentes que conforman el sistema.

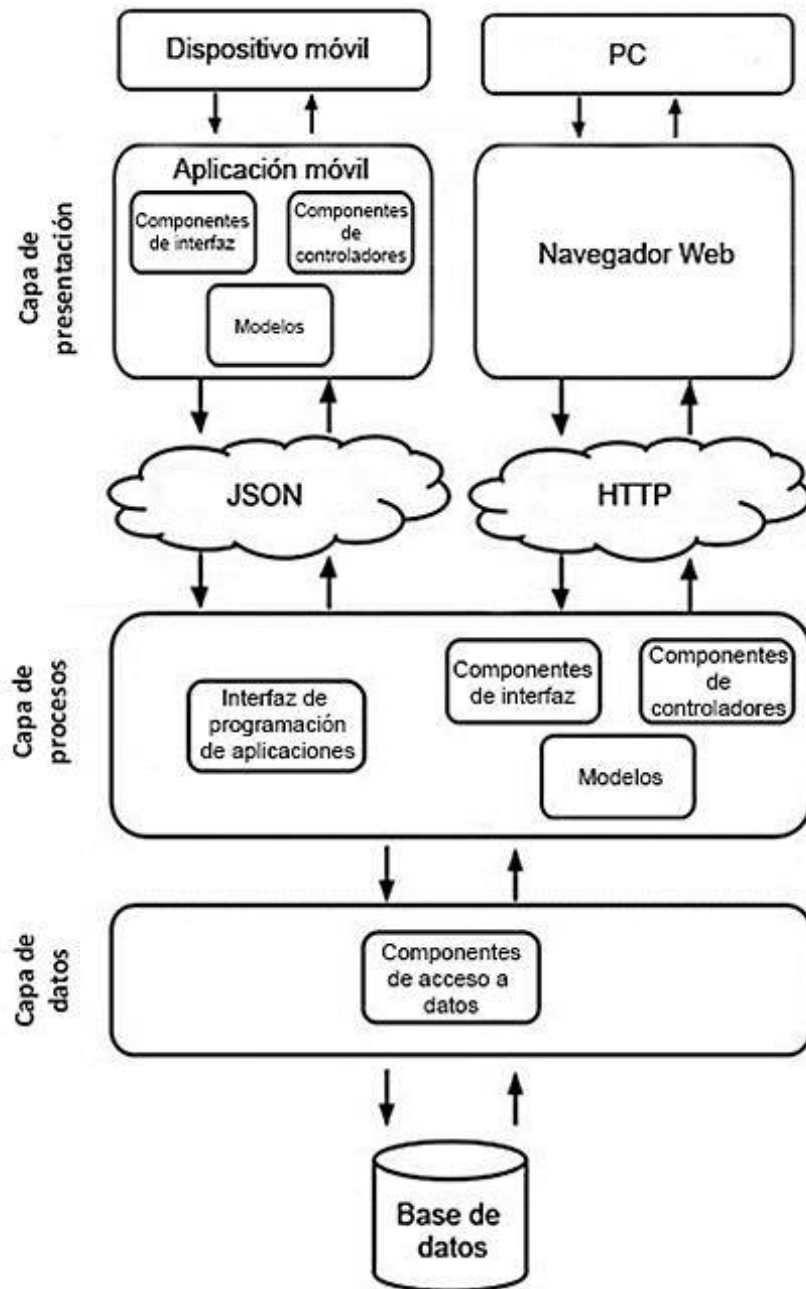


Figura 18. Vista lógica de la arquitectura

Fuente: Elaboración Propia

Bajo esta lógica, se ha definido una arquitectura de 3 capas: capa de presentación, capa de proceso y capa de datos.

- **Capa de presentación**

Está conformada por el navegador web y la aplicación instalada en el dispositivo móvil que se encargaran de mostrar la interfaz final al usuario.

- **Capa de proceso**

Está conformada por los componentes necesarios para construir y organizar los datos que serán mostrados a los usuarios.

- **Capa de datos**

Está conformada por los componentes necesarios para obtener los datos que requiera el sistema.

A continuación se detallara información sobre los componentes del proyecto en cada capa lógica:

- **Capa de presentación**

- **Modelos:** Está representado por las entidades necesarias para el funcionamiento de la aplicación.
- **Componentes de interfaz:** Está representado por todos los recursos necesarios para mostrar la interfaz al usuario.
- **Componentes de controladores:** Está representado por la lógica de la aplicación, es decir todas las clases que implementan los procesos y cálculos necesarios para cumplir con las funcionalidades de la aplicación. Responde a acciones del usuario y se encarga de invocar las peticiones necesarias al Modelo y la Vista.

- **Capa de proceso**

- **Interfaz de programación de aplicaciones:** Se encarga de realizar las consultas necesarias a la base de datos y organizar la información e imprimirla en formato JSON.
- **Componentes de interfaz:** Está representado por las clases usadas para mostrar la interfaz de administración web.
- **Componentes de controladores:** Está representado por la lógica del sistema de administración web.
- **Modelos:** Está representado por las entidades necesarias para el funcionamiento de la aplicación.

- **Capa de datos**

- **Componentes de acceso a datos:** Está representado por el software que extraer información de la base de datos, esta información es entregada a la capa de procesos cada vez que sea requerida por ella.

4.2.3. Validación de la Arquitectura

Los beneficios de poder utilizar la arquitectura cliente – servidor propuesta son los siguientes:

- Centralización de los recursos: los recursos y la integridad de los datos son controlados por el servidor de forma que facilita la tarea de poner al día los datos y recursos.
- Escalabilidad: Se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser mejorado en cualquier momento.
- Interoperabilidad: la incorporación de la tecnología API facilita la integración de nuevos clientes desarrollados en diferentes plataformas.

- Seguridad: Al disponer de un mecanismo central de autenticación, las posibilidades de acceso indebido se reducen considerablemente.

4.2.4. Equipo de Trabajo

Se describen las principales responsabilidades de cada uno de los responsables del desarrollo del proyecto durante las fases de la metodología ágil AUP utilizada; todas las responsabilidades para el desarrollo del presente proyecto, han sido asumidas por los tesistas:

*Tabla 25.
Roles y Responsabilidades*

Rol	Descripción	Responsable
Analista de Sistemas	Realiza actividades de captura de requerimientos y análisis de la solución. Además se encarga de definir la arquitectura de la solución así como el diagrama de clases.	Ramírez López Kelvin Dante
Desarrollador	Realiza la implementación de la solución en base al diagrama de clases definido. Además se encarga de definir el esquema de pruebas para probar las funcionalidades de la solución.	Ramírez López Kelvin Dante
Tester	Realiza la ejecución del esquema de pruebas definido. Esto incluye las pruebas unitarias.	Quiroz Dávila Jefferson Leandro

Fuente: Elaboración propia

4.2.5. Diagrama de Clases de Análisis

Usuario: Contiene la información de todos los usuarios que ingresan al sistema.

Administrador: Contiene la información de los usuarios que tiene acceso a todos los módulos de la página web.

Colaborador: Contiene la información de los usuarios que tiene acceso a ciertos módulos de la página web.

Movil: Contiene la información de los usuarios que acceden sin login al aplicativo móvil.

Nodo: Contiene la información de todos los nodos que conforman el grafo.

Destino: Contiene la información de los destinos que son objeto de búsqueda en el grafo.

Arista: Contiene la identificación de los nodos que guardan una relación bidireccional entre sí.

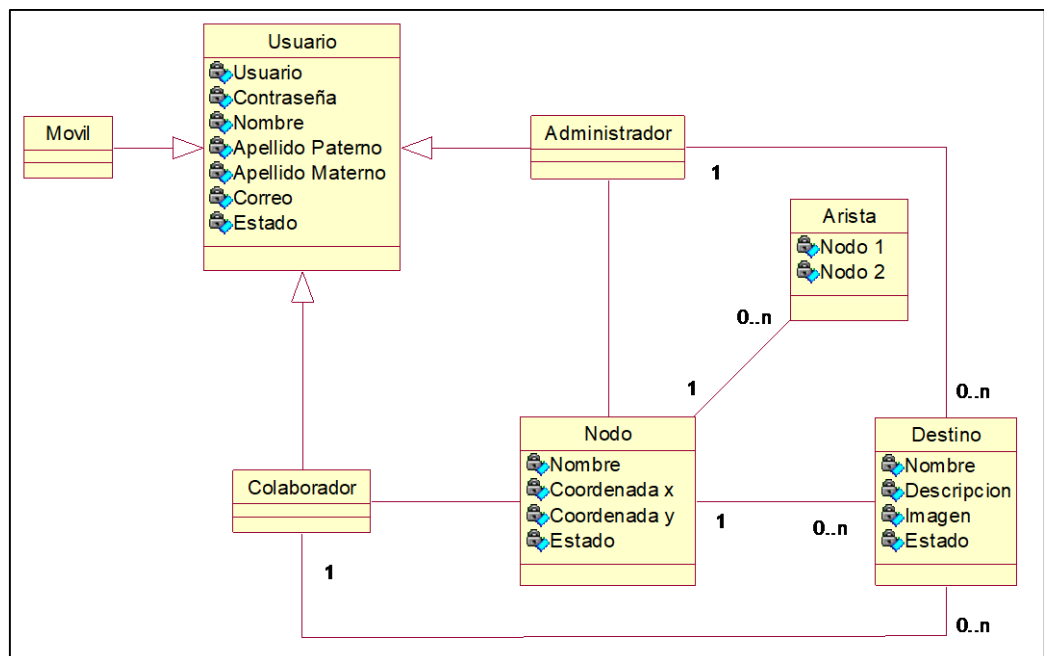


Figura 19. Diagrama de Clases de Análisis

Fuente: Elaboración Propia

4.3. CONSTRUCCIÓN

Situados en esta parte del proceso de desarrollo de software, cabe indicar que el objetivo es construir el software trabajando en forma iterativa e incremental, de acuerdo a los requerimientos capturados en la fase inicial.

A continuación, tomaremos como tarea inicial el modelamiento del sistema y posteriormente la construcción del sistema a través del diseño de las interfaces:

4.3.1. Modelamiento del Sistema

El objeto del modelado del sistema es entender la lógica de negocio de la aplicación, el dominio del problema del proyecto e identificar una solución viable para el dominio del problema. En nuestro modelado tenemos a los siguientes diagramas: Diagrama de paquetes, diagrama de clases de diseño, diagrama de componentes, diagrama de despliegue, diagramas de secuencia y el diagrama de base de datos.

4.3.1.1. Diagrama de paquetes

El diagrama de paquetes sirve para organizar los elementos de un sistema en un grupo; es decir, mostrar que ciertas clases o componentes son parte de un subsistema en particular como lo muestra la Figura 20.

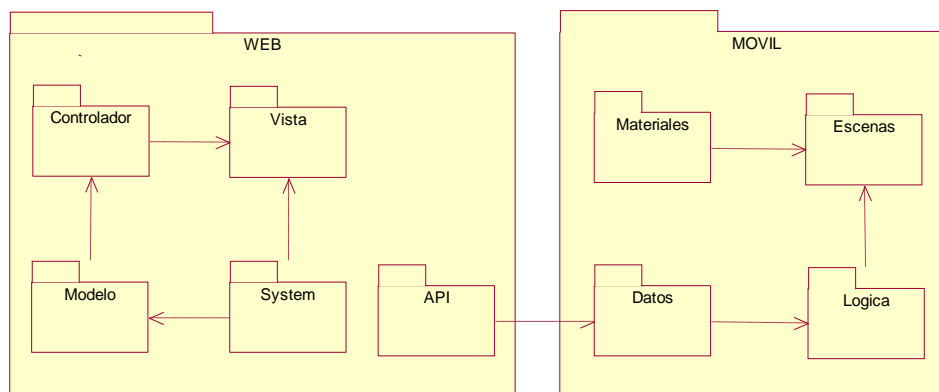


Figura 20. Diagrama de Paquetes

Fuente: Elaboración Propia

Paquete Vista: paquete que agrupa las interfaces HTML5 de la página web.

Paquete Controlador: Paquete donde se crea la lógica del negocio de la página web en scripts PHP.

Paquete Modelo: Paquete donde se realizan las instrucciones SQL (Inserción, Actualización, Eliminación y Consultas) de la página web.

Paquete System: Paquete que reúne funcionalidades del sistema como: conexión de los servicios a la base de datos, frameworks de estilos e imágenes.

Paquete API: Paquete donde se crea la interfaz de programación de aplicaciones que permite la disposición de los datos a la aplicación móvil.

Paquete Datos: Paquete donde se realiza la captura de los datos de la interfaz de programación de aplicaciones.

Paquete lógica: Paquete que almacena los scripts C# que contiene la lógica del negocio de la aplicación móvil.

Paquete Scenes: Paquete que agrupa las escena graficas de la aplicación móvil.

Paquete Materiales: Paquete que agrupa los objetos 2D y 3D de la aplicación móvil.

4.3.1.2. Diagrama de clases de diseño

El diagrama de clases de diseño que se observa en la Figura 21, muestra a las entidades de nuestro sistema con sus respectivos atributos (propiedades) con las acciones que realizan.

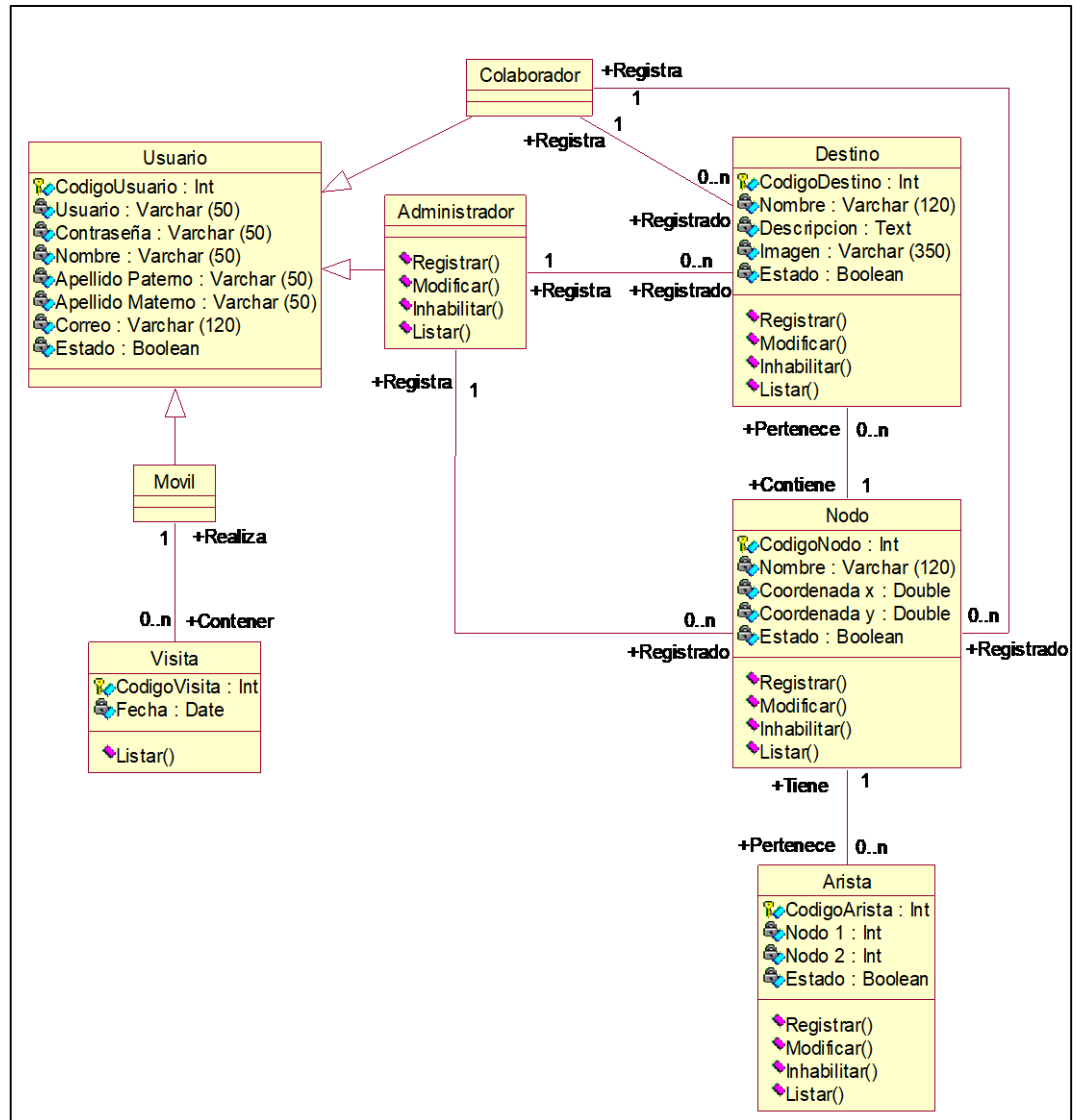


Figura 21. Diagrama de Clases de Diseño

Fuente: Elaboración Propia

4.3.1.3. Diagrama de componentes

Un diagrama de componentes representa a un elemento real (un componente de software). La Figura 22 muestra a los componentes de nuestro software relacionados entre sí.

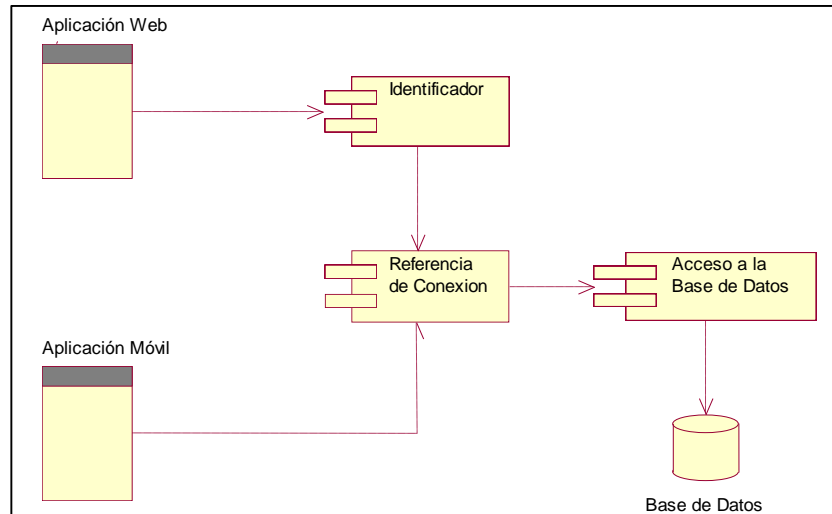


Figura 22. Diagrama de Componentes

Fuente: Elaboración Propia

4.3.1.4. Diagrama de despliegue

Conocidos también como diagramas de distribución cuya función es mostrar la arquitectura física de nuestro sistema; para tal caso, en la Figura 23 podemos observar la distribución de los componentes que lo integran.

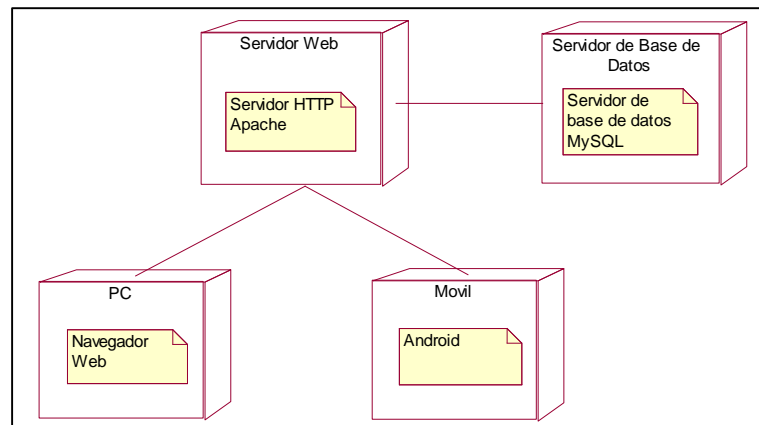


Figura 23. Diagrama de Despliegue

Fuente: Elaboración Propia

4.3.1.5. Diagrama de secuencia

Iniciar sesión

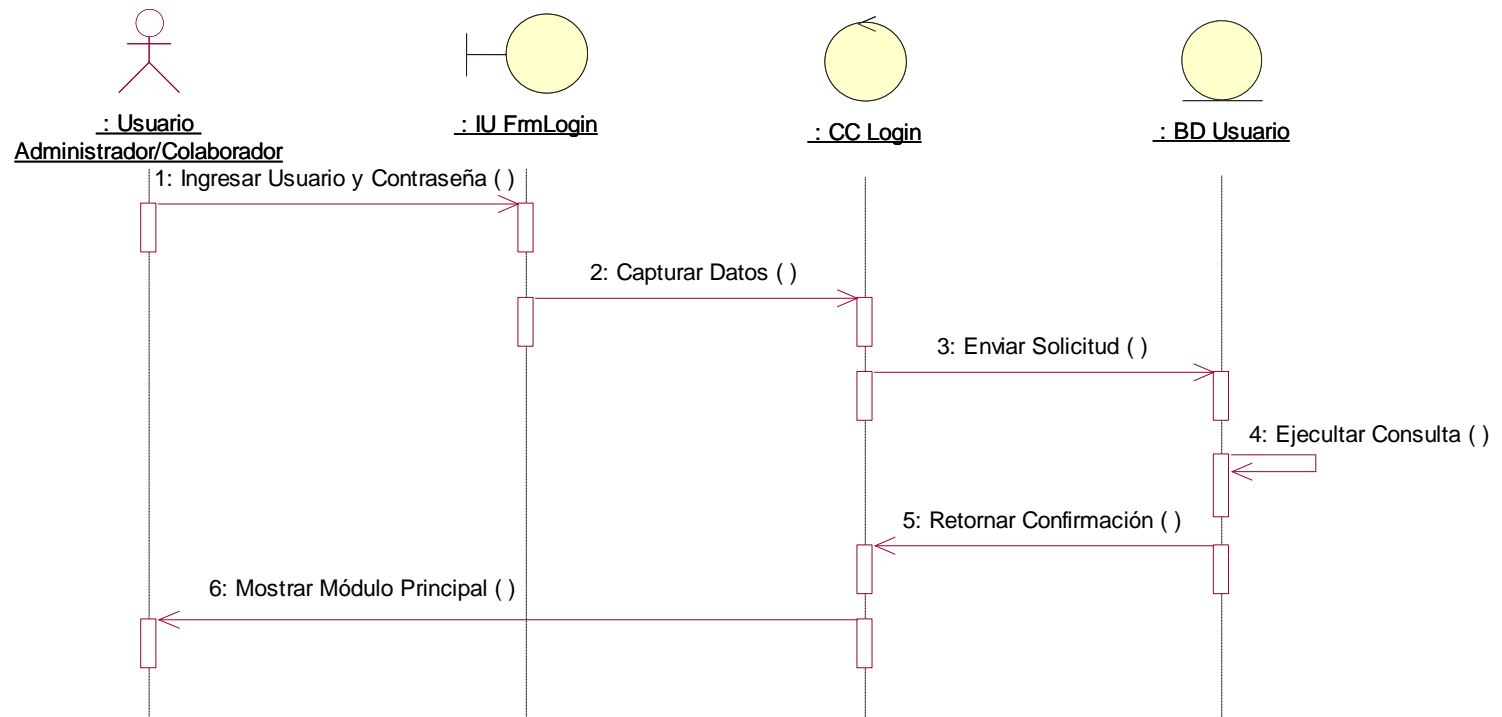


Figura 24. Diagrama de Secuencia - Iniciar Sesión
Fuente: Elaboración Propia

Gestionar Usuario

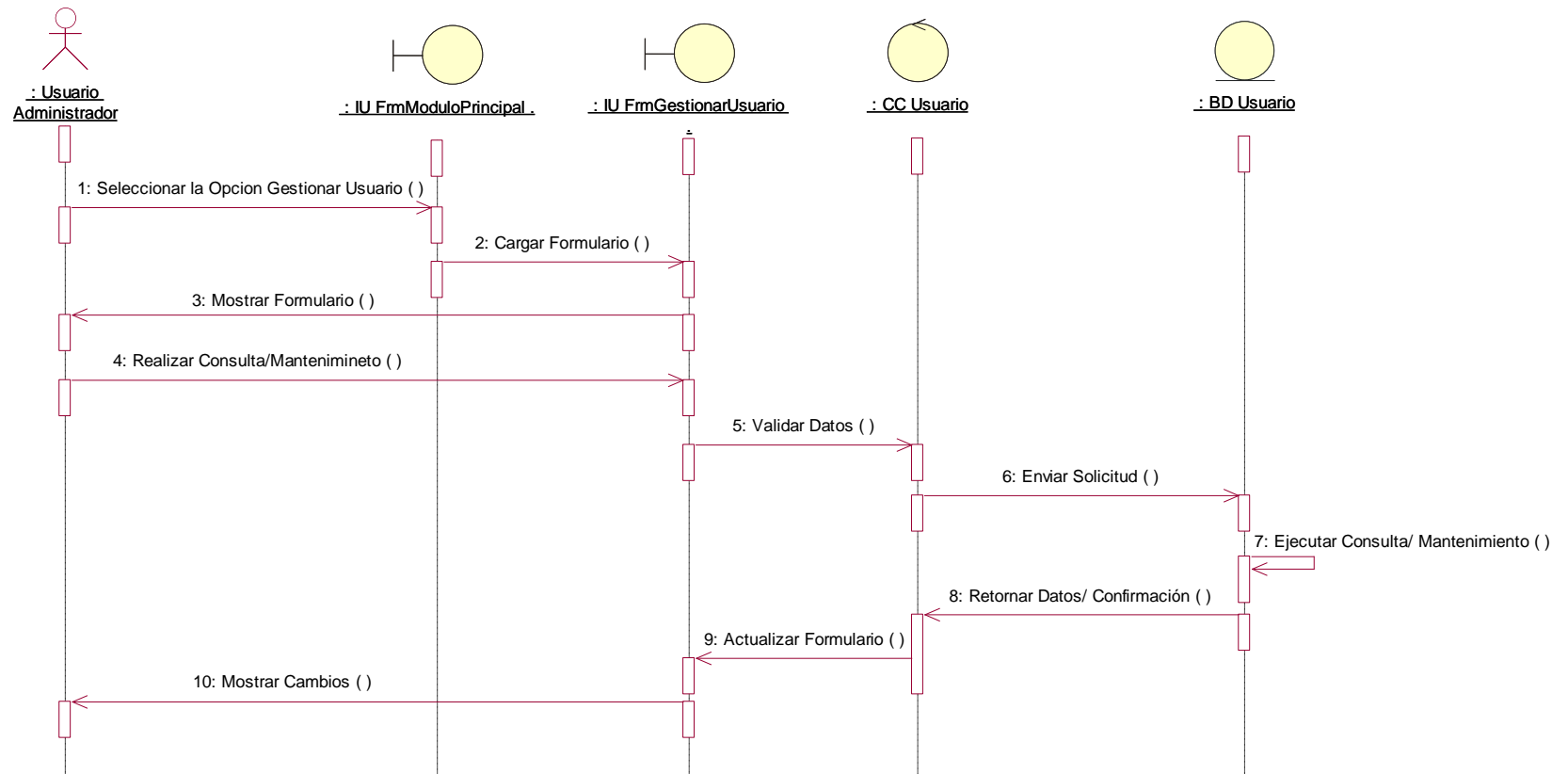


Figura 25. Diagrama de Secuencia - Gestionar Usuario
Fuente: Elaboración Propia

Gestionar Nodo

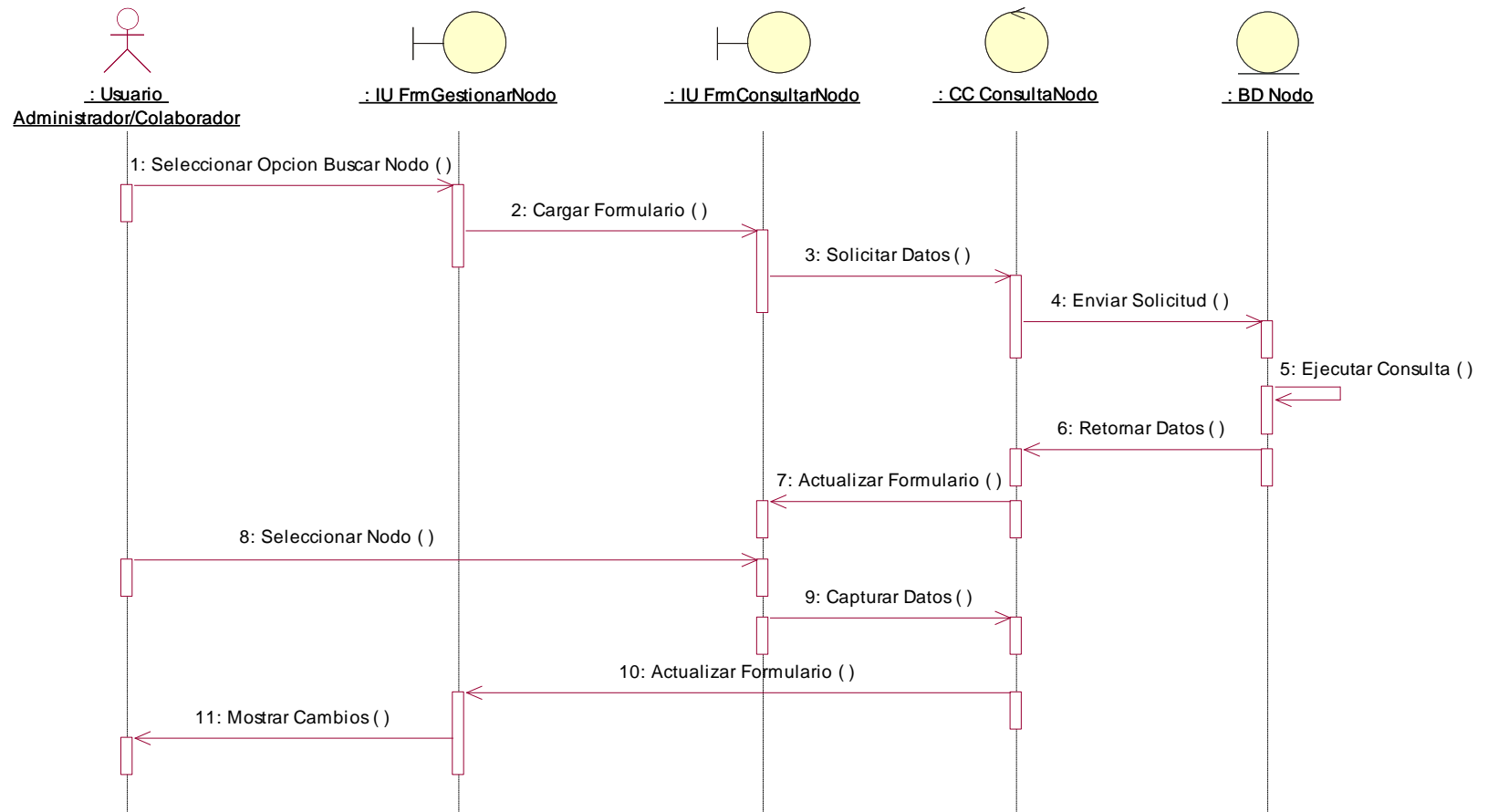


Figura 26. Diagrama de Secuencia - Gestionar Nodo
Fuente: Elaboración Propia

Gestionar Arista

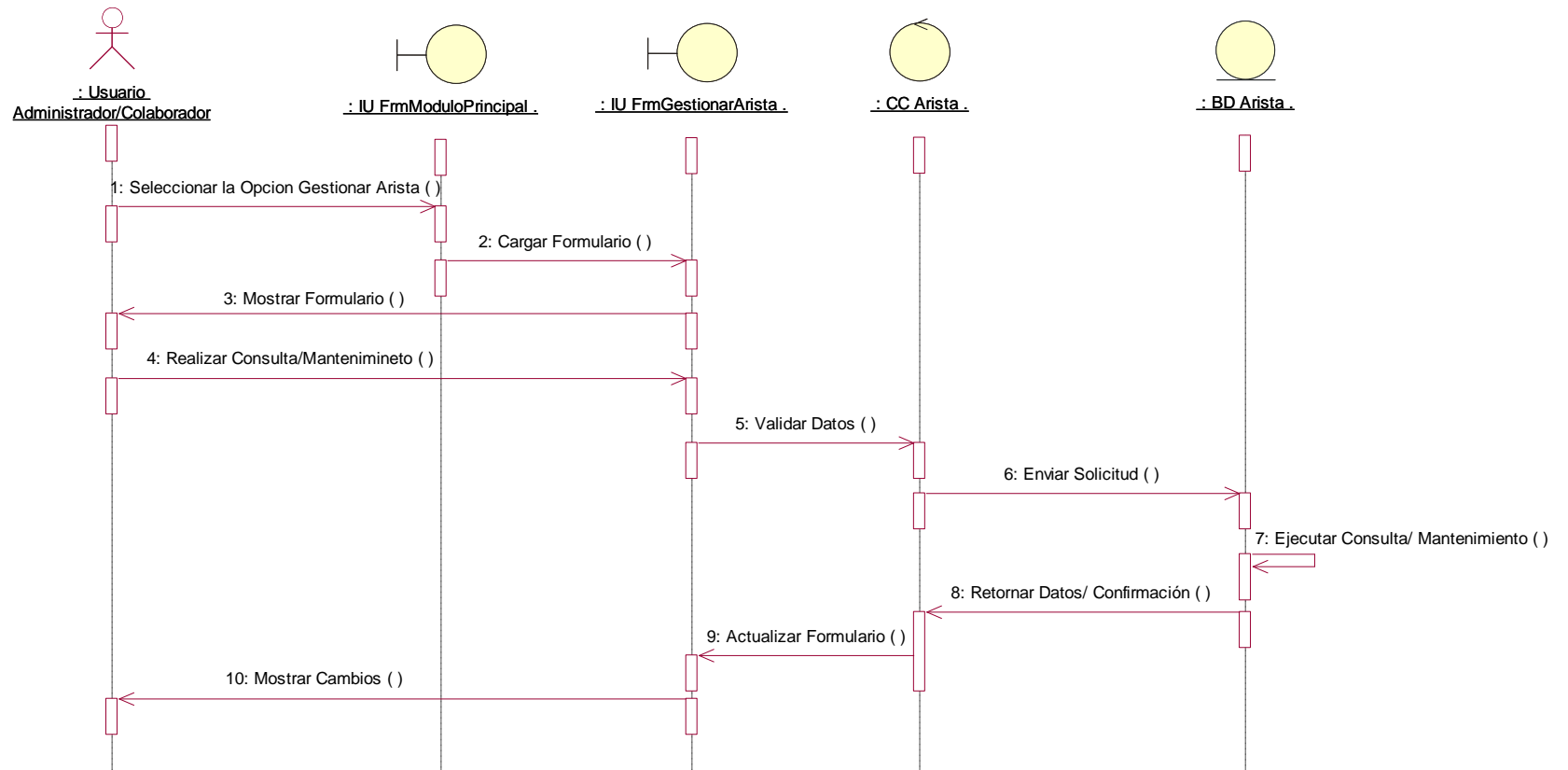


Figura 27. Diagrama de Secuencia - Gestionar Arista
Fuente: Elaboración Propia

Gestionar Destino

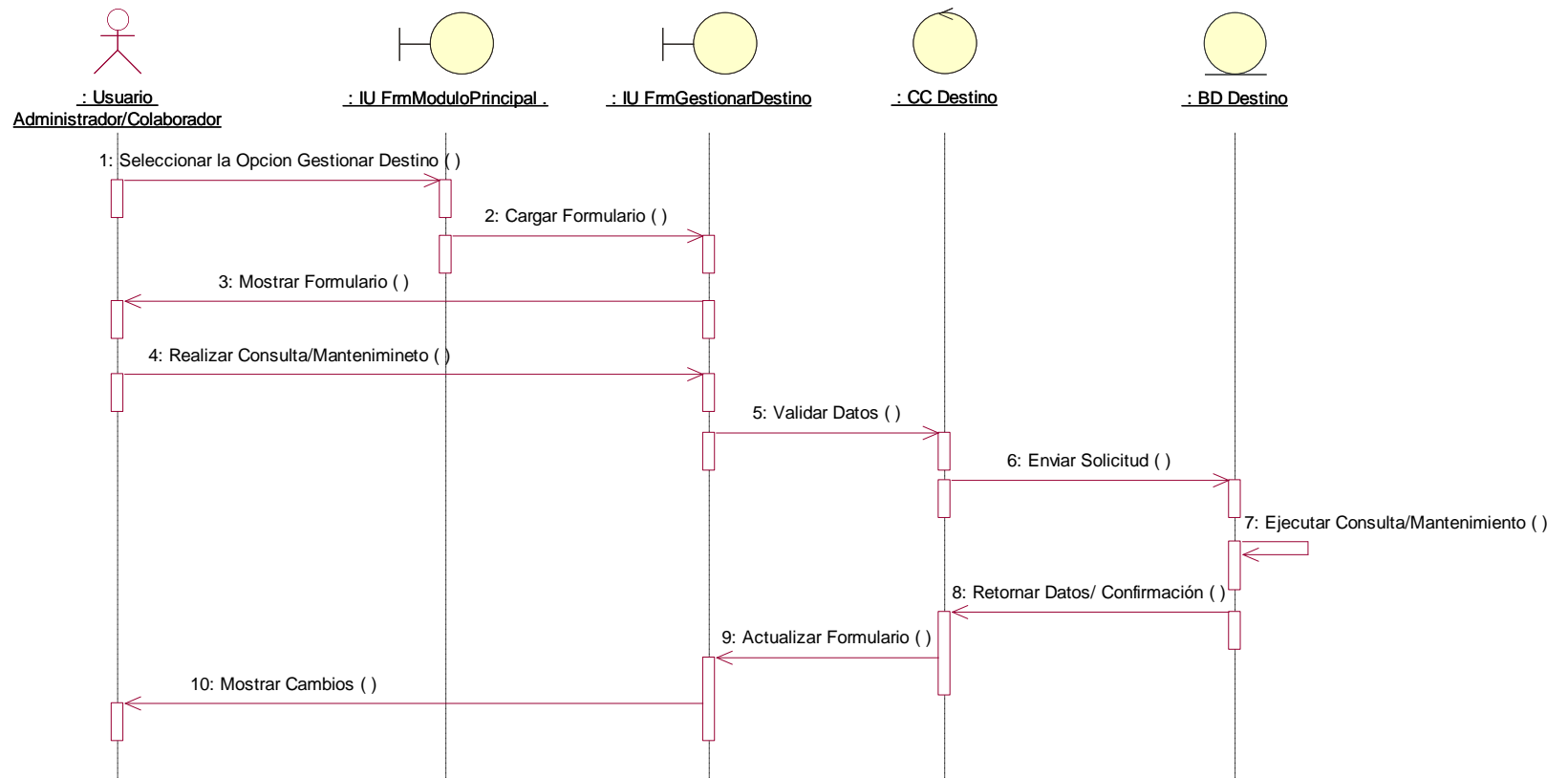


Figura 28. Diagrama de Secuencia – Gestionar Destino
Fuente: Elaboración Propia

Generar Grafo

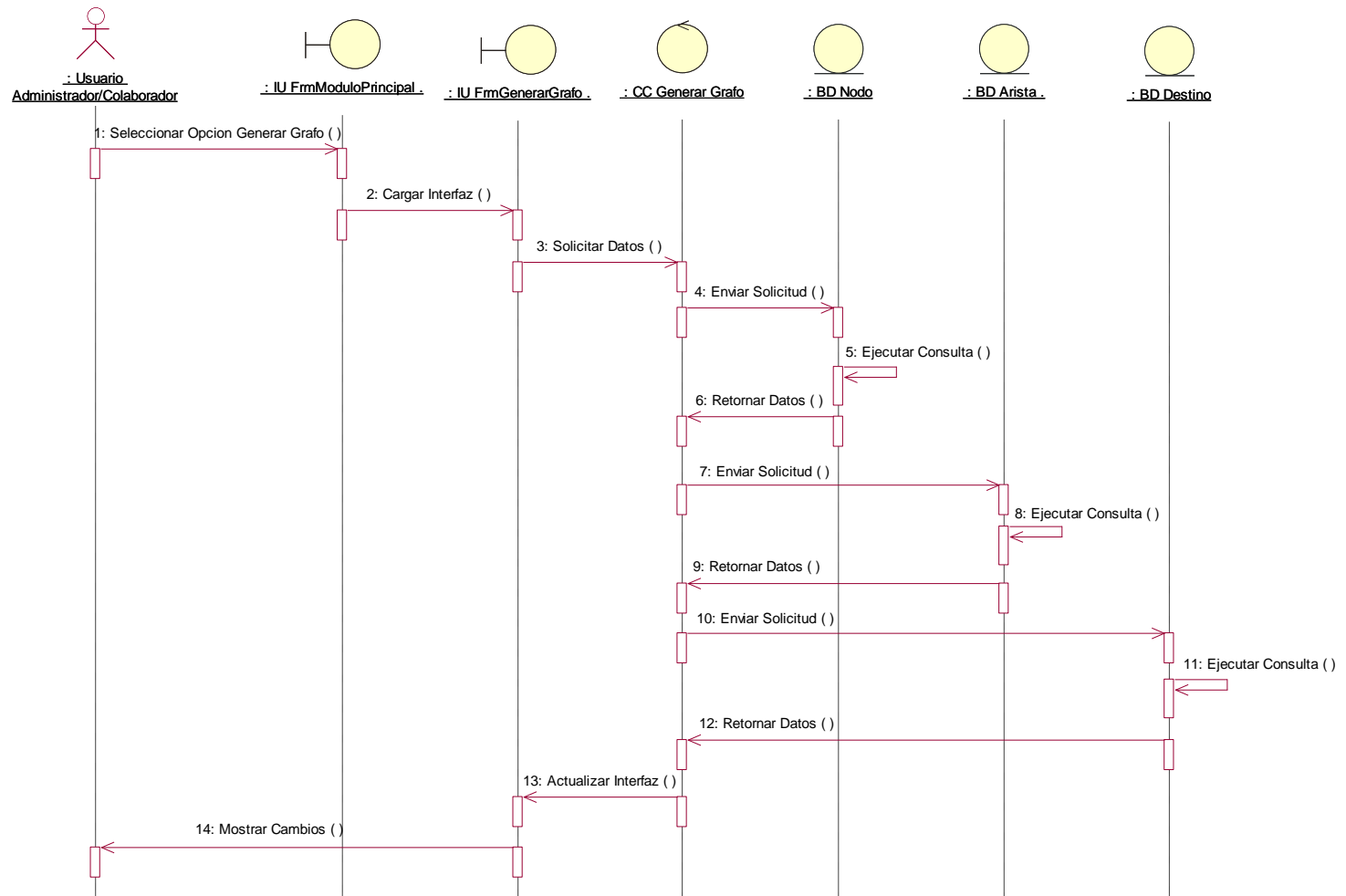


Figura 29. Diagrama de Secuencia – Generar Grafo
Fuente: Elaboración Propia

Generar Reporte Visitantes

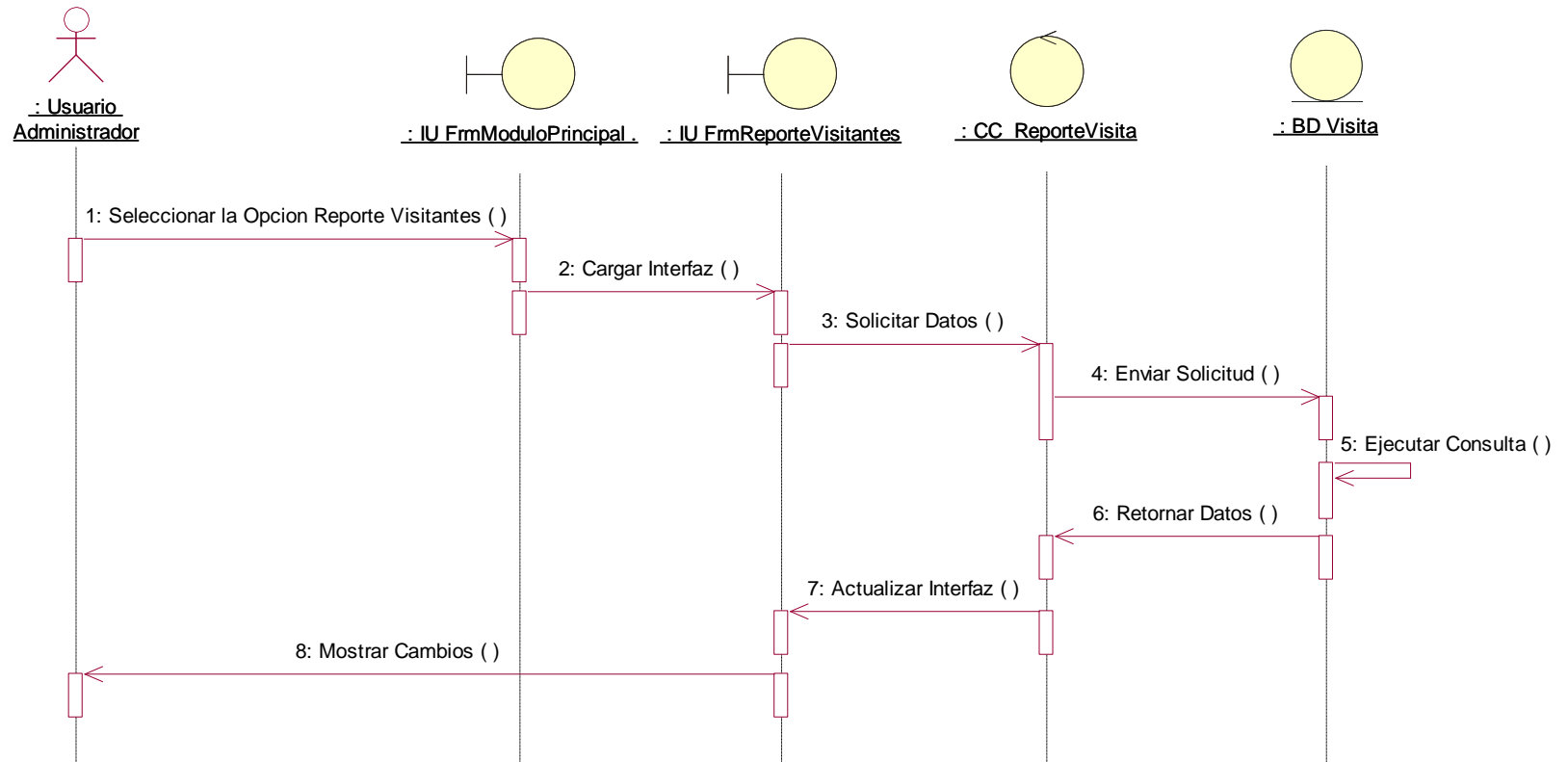


Figura 30. Diagrama de Secuencia – Generar Reporte Visitantes
Fuente: Elaboración Propia

Cargar Datos

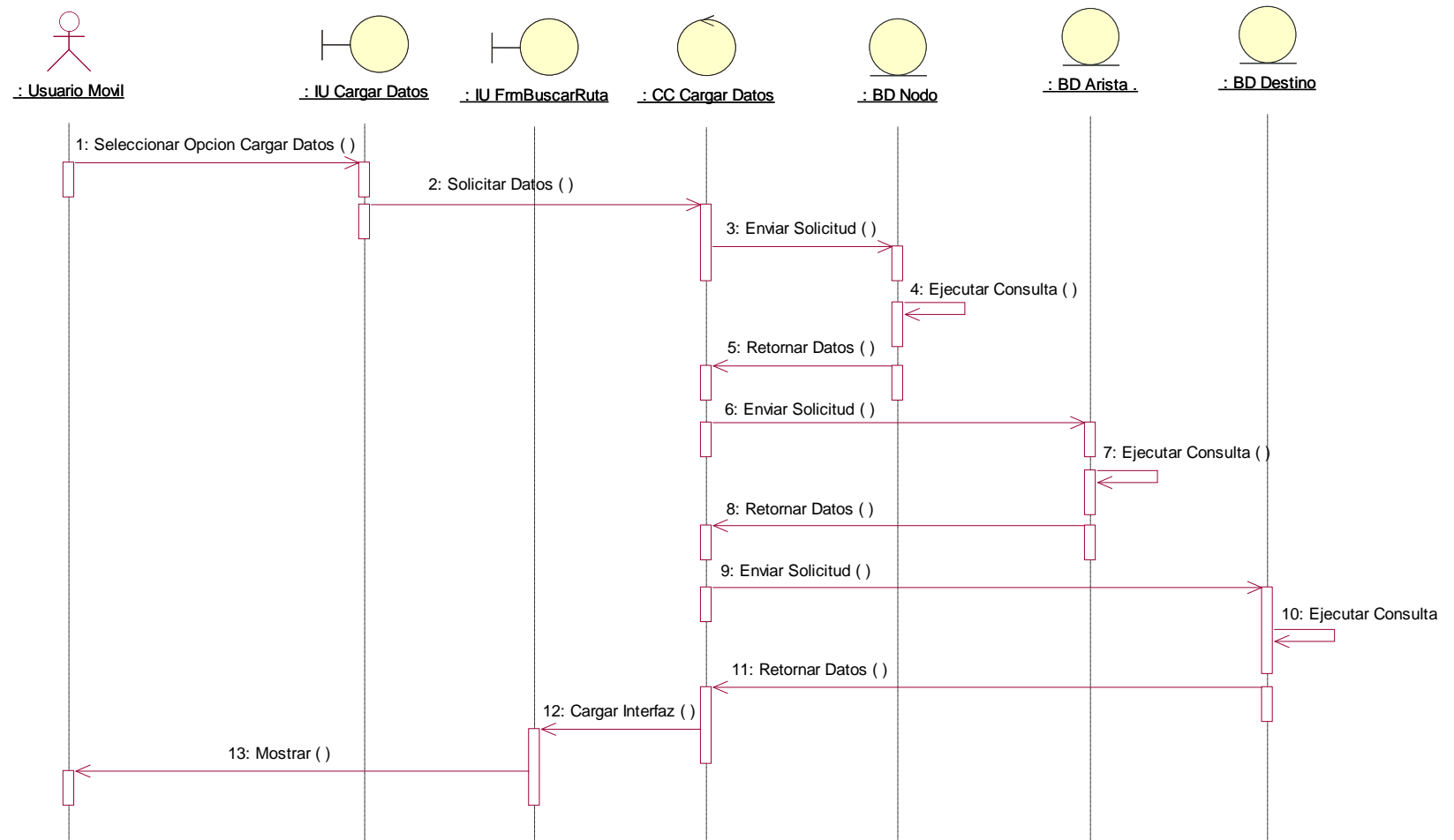


Figura 31. Diagrama de Secuencia - Cargar Datos
Fuente: Elaboración Propia

Consultar Origen

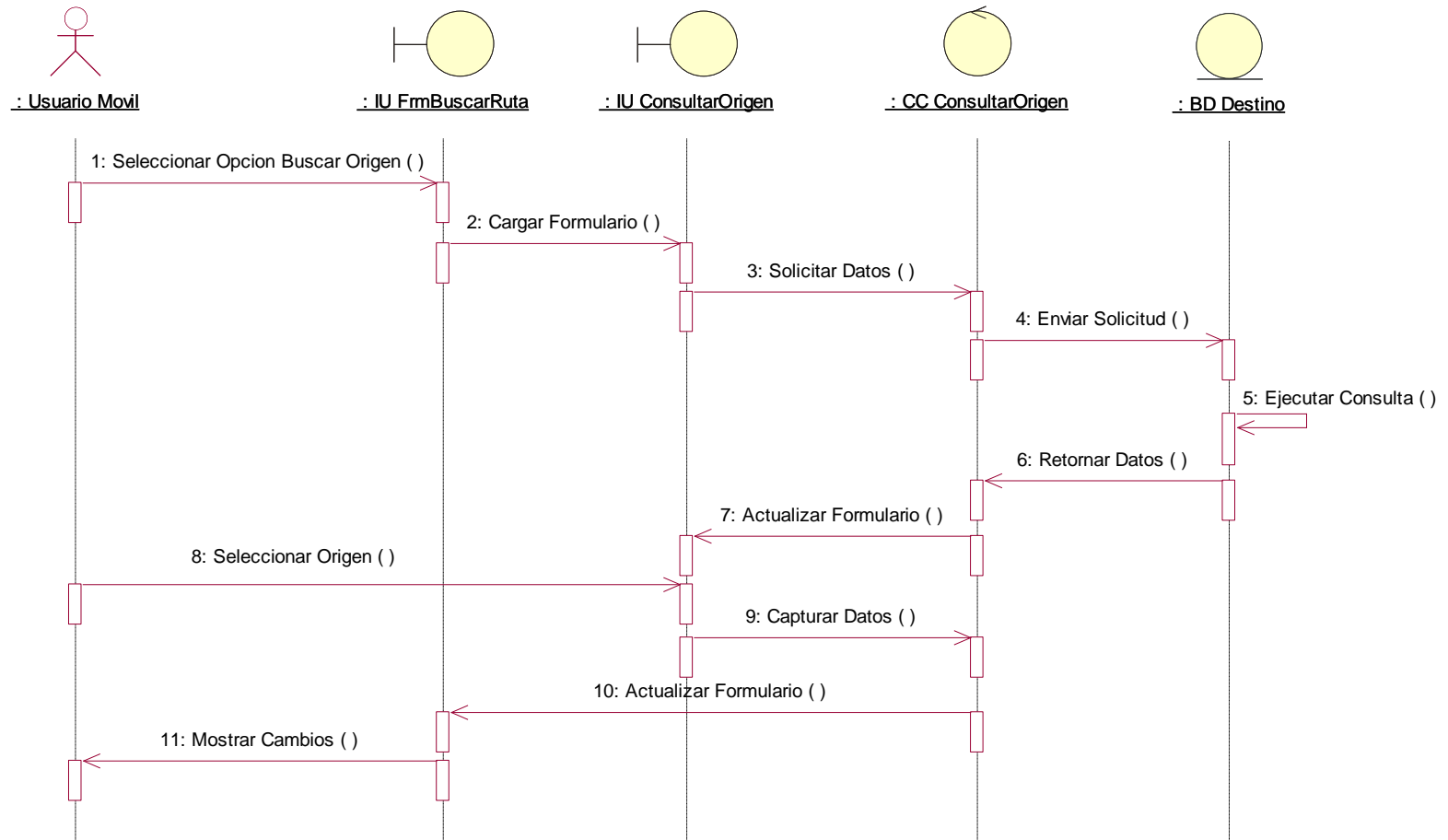


Figura 32. Diagrama de Secuencia - Consultar Origen
Fuente: Elaboración Propia

Consultar Destino

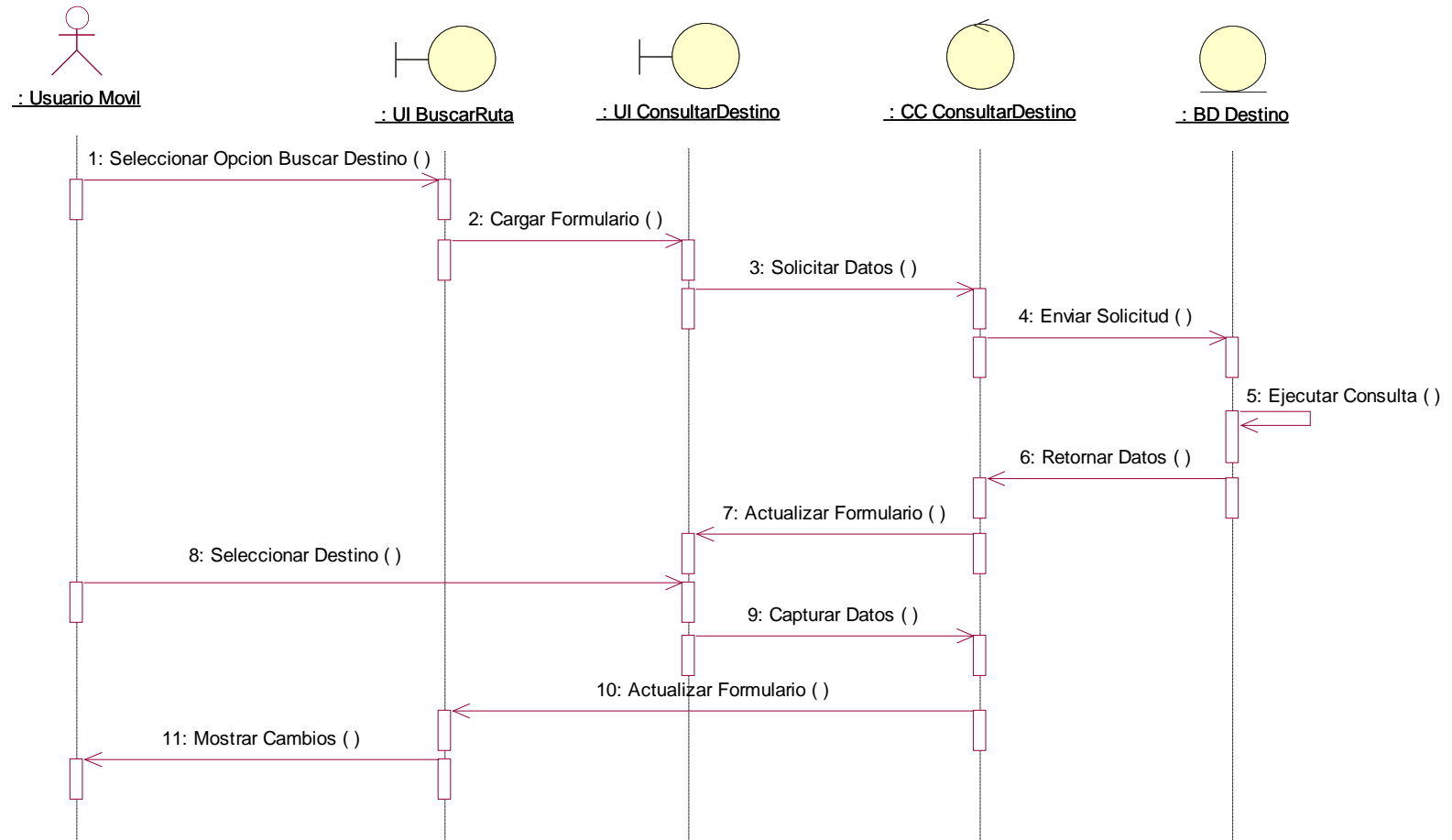


Figura 33. Diagrama de Secuencia - Consultar Destino
Fuente: Elaboración Propia

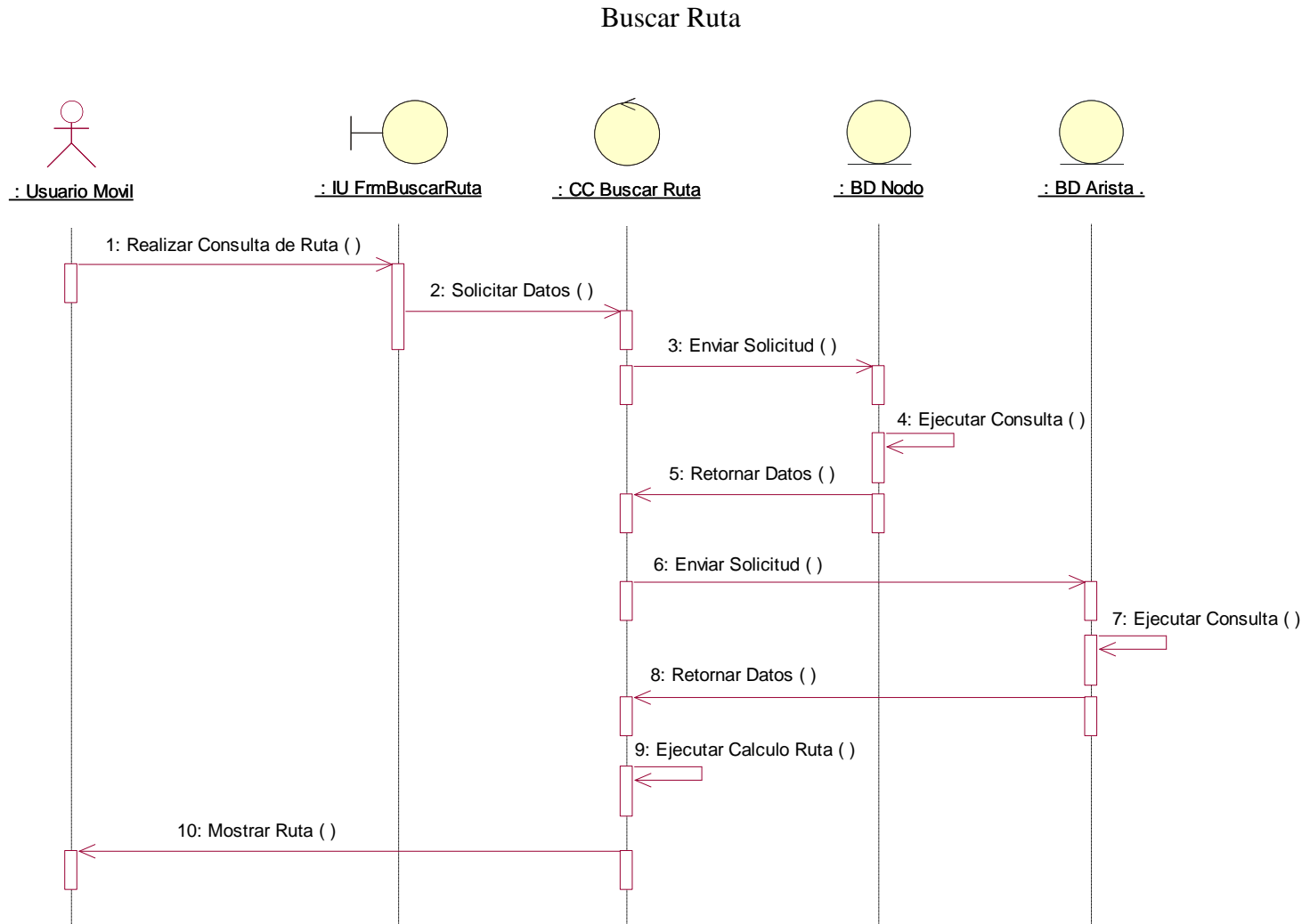


Figura 34. Diagrama de Secuencia - Buscar Ruta
Fuente: Elaboración Propia

4.3.1.6. Diagrama de Base de Datos

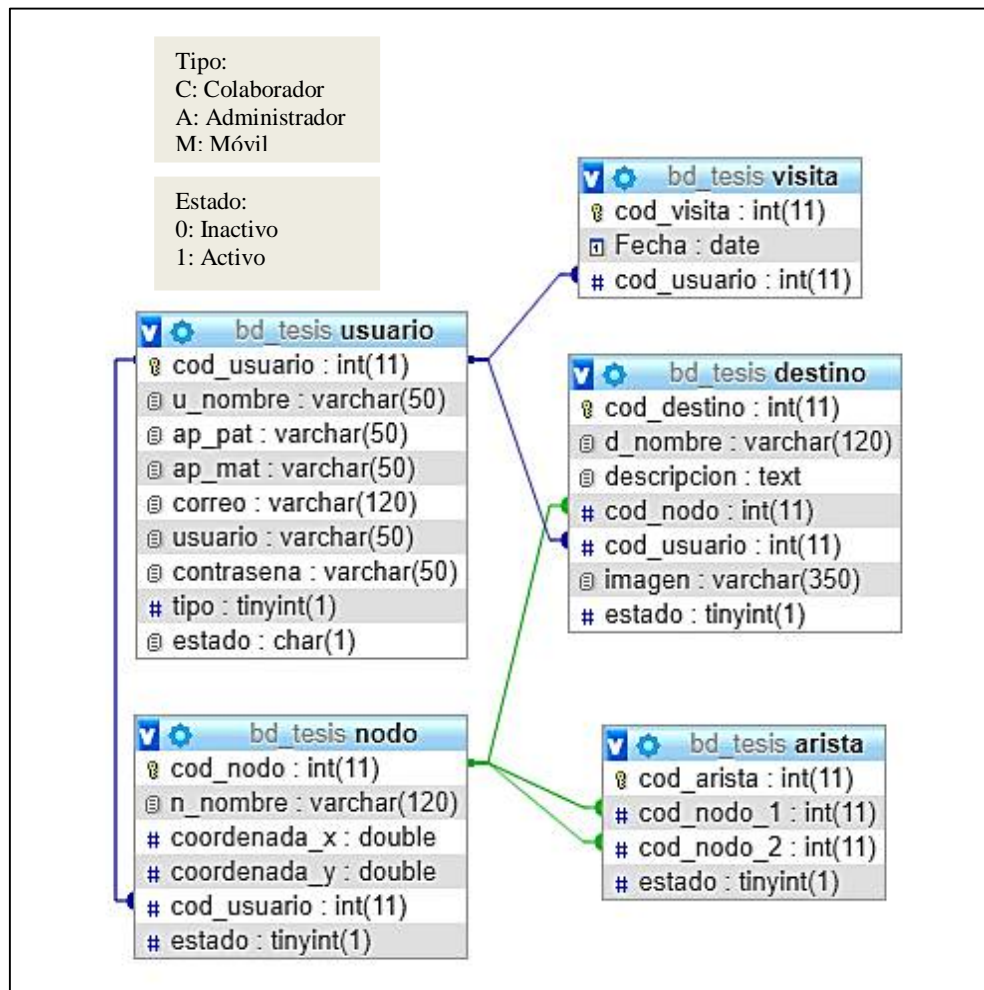


Figura 35. Diagrama de Base de Datos
Fuente: Elaboración Propia

4.3.1.7. Diseño de interfaces

Módulo de Usuario:

Cargar Datos:

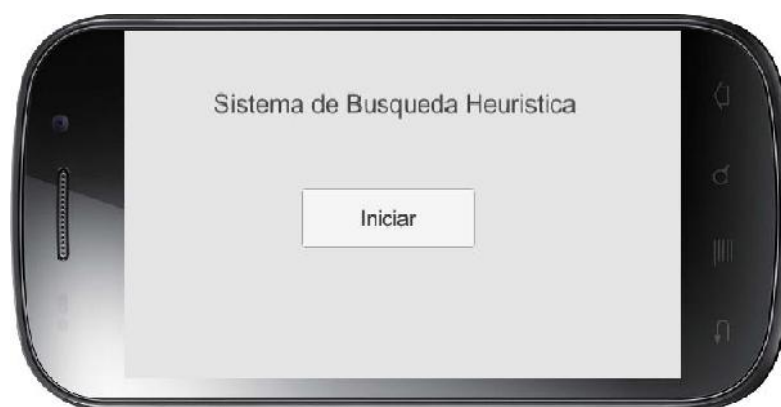


Figura 36. Interfaz Gráfica – Cargar Datos
Fuente: Elaboración Propia

Buscar Ruta:



Figura 37. Interfaz Gráfica - Buscar Ruta
Fuente: Elaboración Propia

Consultar Origen:



Figura 38. Interfaz Gráfica - Consultar Origen
Fuente: Elaboración Propia

Consultar Destino:

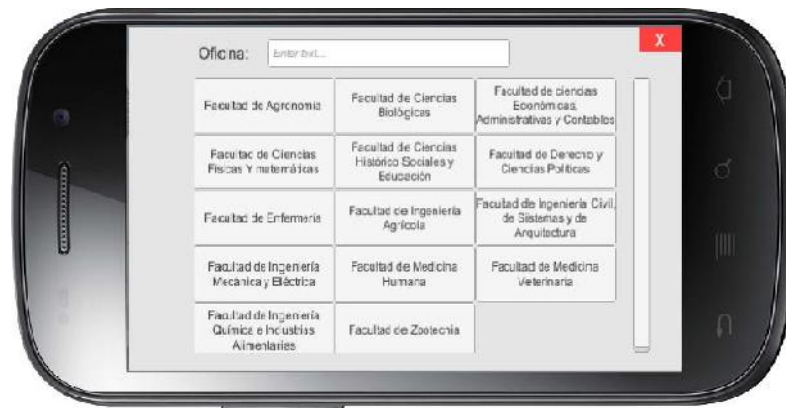


Figura 39. Interfaz Gráfica - Consultar Destino
Fuente: Elaboración Propia

Módulo de Administración:

Iniciar Sesión:

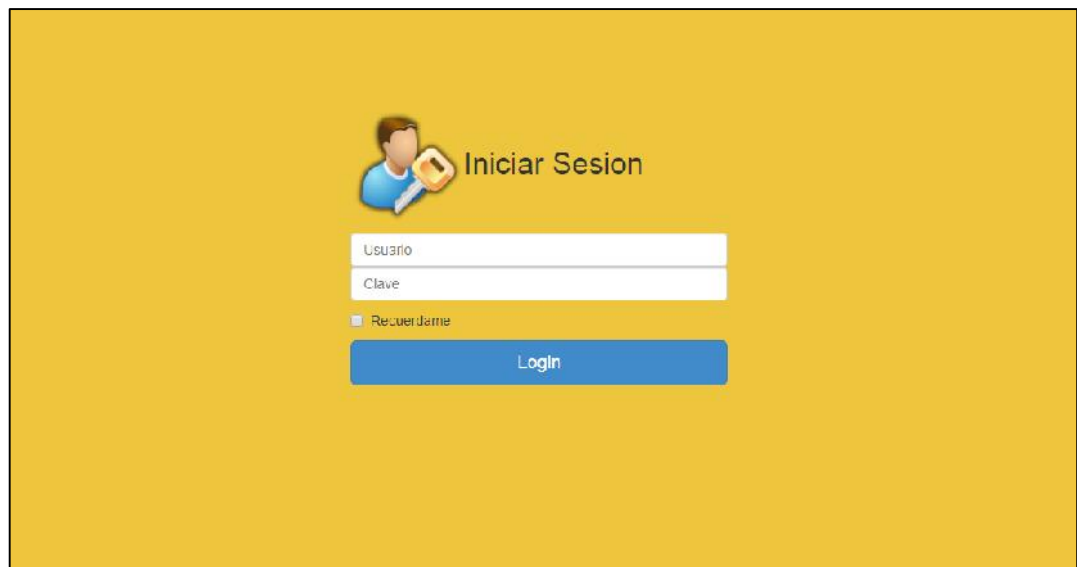


Figura 40. Interfaz Gráfica - Iniciar Sesión
Fuente: Elaboración Propia

Gestionar Usuario:

Inicio

Usuario

Nodo

Destino

Arista

Grafo

Cerrar Sesión

LISTADO DE USUARIOS

NOMBRE COMPLETO	CORREO	USUARIO	CONTRASEÑA	ESTADO	ACCION
Administrador Administrador Administrador	Ejemplo@gmail.com	Admin	Admin	1	<div></div> <div></div>
Bienvenido(a) Administrador 1 Administrador					

Figura 41. Interfaz Gráfica - Gestionar Usuario
Fuente: Elaboración Propia

Gestionar Nodo:

	Inicio	Usuario	Nodo	Destino	Arista	Grafo	Cerrar Sesión
LISTADO DE NODOS							
NOMBRE	COORDENADA X	COORDENADA Y	USUARIO	ESTADO	ACCION		
nodo 1	-8.8	1.74	Administrador	1	 		
nodo 2	-8.1	20.5	Administrador	1	 		
nodo 3	-5.9	81.68	Administrador	1	 		
nodo 4	-19.3	20.45	Administrador	1	 		
nodo 5	2.95	25.8	Administrador	1	 		
nodo 6	6.68	25.48	Administrador	1	 		
nodo 7	23.6	76.28	Administrador	1	 		
nodo 8	27.61	76.94	Administrador	1	 		
nodo 9	2.38	20.45	Administrador	1	 		
nodo 10	-4	104.39	Administrador	1	 		
nodo 11	1	81.73	Administrador	1	 		

Figura 42. Interfaz Gráfica- Gestionar Nodo
Fuente: Elaboración Propia

Gestionar Destino

Inicio	Usuario	Nodo	Destino	Arista	Grafo	Cerrar Sesión
LISTADO DE DESTINOS						+
NOMBRE	DESCRIPCION	NODO	USUARIO	IMAGEN	ESTADO	ACCION
Facultad de Agronomía	Hora de Atención: 8 am - 1:30 pm Ubicación: segundo piso	nodo 204	Administrador		1	 
Facultad de Ciencias Biológicas	Hora de Atención: 8 am - 1:30 pm Ubicación: segundo piso	nodo 199	Administrador		1	 
Facultad de ciencias Económicas, Administrativas y Contables	Hora de Atención: 8 am - 1:30 pm Ubicación: primer piso	nodo 201	Administrador		1	 
Facultad de Ciencias Físicas Y matemáticas	Hora de Atención: 8 am - 1:30 pm Ubicación: primer piso	nodo 150	Administrador		1	 
Facultad de Ciencias Histórico Sociales y Educación	Hora de Atención: 8 am - 1:30 pm Ubicación: primer piso	nodo 252	Administrador		1	 
Facultad de Derecho y Ciencias Políticas	Hora de Atención: 8 am - 1:30 pm	nodo	Administrador		1	 

Figura 43. Interfaz Gráfica - Gestionar Destino

Fuente: Elaboración Propia

Gestionar Arista

























Inicio	Usuario	Nodo	Destino	Arista	Grafo	Cerrar Sesión
LISTADO DE ARISTAS						+
ARISTA	NODO 1	NODO 2	ACCION			
1	nodo 1	nodo 42				
2	nodo 1	nodo 23				
3	nodo 1	nodo 2				
4	nodo 2	nodo 1				
5	nodo 2	nodo 4				
6	nodo 2	nodo 9				
7	nodo 2	nodo 51				
8	nodo 3	nodo 43				
9	nodo 3	nodo 12				
10	nodo 3	nodo 11				
11	nodo 3	nodo 10				
12	nodo 4	nodo 42				

Figura 44. Interfaz Gráfica - Consultar Nodo

Fuente: Elaboración Propia

Generar Grafo



Figura 45. Interfaz Gráfica- Generar grafo
Fuente: Elaboración Propia

Generar Reporte Visitantes

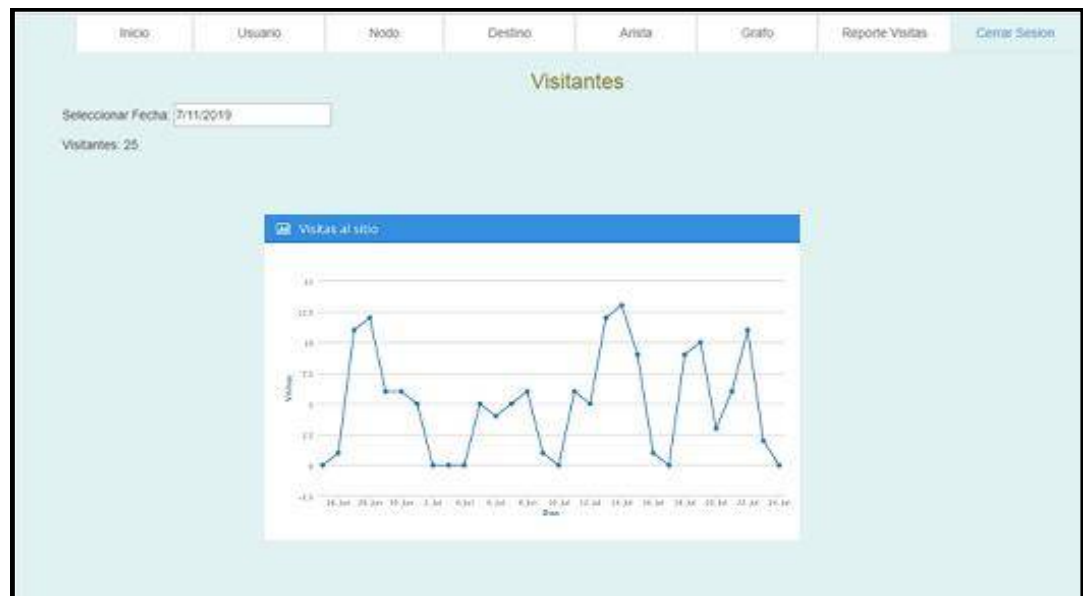


Figura 46. Interfaz Gráfica- Generar Reporte Visitantes
Fuente: Elaboración Propia

4.3.2. Construcción del Sistema

4.3.2.1. Requerimientos en hardware y software

Los requerimientos necesarios para la ejecución del sistema de búsqueda heurística son los siguientes:

Hardware:

- Computadora personal:
 - Memoria RAM: 2GB
 - Almacenamiento: 80 GB
 - Procesador: Intel corel dos duo
- Dispositivo Android:
 - Memoria RAM: 1 Gb
 - Almacenamiento: 4 Gb
 - Procesador: Qualcomm Snapdragon 660
- Servidor Web:
 - Memoria RAM: 16GB
 - Almacenamiento: 500 GB
 - Procesador: Intel Xeon inside

Software:

- Computadora personal:
 - Sistema Operativo: Windows 7 Professional
 - Navegador: Google chrome (Versión 75.0.3770.142)
- Dispositivo Android:
 - Sistema Operativo: Android 4.4
- Servidor Web:
 - Sistema Operativo: Centos 6

4.3.2.2. Algoritmo A Star para el Sistema de búsqueda heurística

El código correspondiente al algoritmo A Star se realizará en la codificación de la aplicación móvil y es el siguiente:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Buscador : MonoBehaviour {
    public bool activar;
    public List<NodoEspecial> caminoTotal;
    public List<NodoEspecial> caminoFinal;
    public GameObject inicio;
    public GameObject final;
    public int pasoActual;
    private LineRenderer Linea;
    // Use this for initialization
    void Start () {
        this.activar = false;
        this.pasoActual = 0;
        Linea = GetComponent<LineRenderer> ();
    }

    // Update is called once per frame
    void Update () {
        if(this.activar){
            this.caminoTotal = new List<NodoEspecial>();
            this.caminoFinal = new List<NodoEspecial>();
            this.pasoActual = 0;
            AlgoritmoEstrella();
            this.activar = false;
        }
    }

    public void AlgoritmoEstrella(){
        NodoEspecial nuevo = new NodoEspecial(this.inicio, 0,
        false, 99999);
        this.caminoTotal.Add(nuevo);
        Enjambre();
        for(int i = 0; i < this.caminoTotal.Count; i++){
            if(this.caminoTotal[i].actual){
                this.caminoFinal.Add(this.caminoTotal[i]);
            }
        }
        Linea.positionCount = this.caminoFinal.Count;
        for(int i = 0; i < this.caminoFinal.Count; i++){
            Linea.SetPosition (i, this.caminoFinal
            [i].nodo.transform.position);
        }
    }

    public void Enjambre(){
        float minor = this.caminoTotal[0].heuris;
        int location = 0;
        for(int i = 0; i < this.caminoTotal.Count; i++){
            if(this.caminoTotal[i].heuris <= minor){
                minor = this.caminoTotal[i].heuris;
                location = i;
            }
        }
    }
}
```

```

    }
}
if(this.caminoTotal[location].nodo == this.final){
    this.caminoTotal[location].actual = true;
    Debug.Log("Done");
    return;
}
if(this.caminoTotal.Count > 999){
    Debug.Log("Tope de emergencia");
    return;
}
Evaluar(location);
Enjambre();
}

public void Evaluar(int locate){
    if(this.caminoTotal[locate].paso == this.pasoActual){
    }else{
        float wminor = 99999;
        for(int i = 0; i < this.caminoTotal.Count; i++){
            if(this.caminoTotal[i].paso == this.pasoActual){
                if(this.caminoTotal[i].heuris < wminor){
                    wminor = this.caminoTotal[i].heuris;
                }
            }
        }
        this.pasoActual = this.caminoTotal[locate].paso;
        while(this.caminoTotal[this.caminoTotal.Count-1].paso >
            this.caminoTotal[locate].paso){
            this.caminoTotal.RemoveAt(this.caminoTotal.Count-1);
        }
        for(int i = 0; i < this.caminoTotal.Count; i++){
            if(this.caminoTotal[i].paso == this.pasoActual){
                if(this.caminoTotal[i].actual){
                    this.caminoTotal[i].heuris = wminor;
                    this.caminoTotal[i].actual = false;
                }
            }
        }
        Debug.Log("Regresar");
    }
    this.caminoTotal[locate].actual = true;
    this.pasoActual += 1;
    AbrirNodo(locate);
}

public void AbrirNodo(int locate){
    for(int i = 0; i < this.caminoTotal[locate]
        .nodo.GetComponent<NodoV2>().NodeCollection.Length; i++){
        NodoEspecial extra = newNodoEspecial
            (this.caminoTotal[locate].nodo. GetComponent
            <NodoV2>().NodeCollection[i],this.pasoActual,
            false,MedirHeuristica(this.caminoTotal[locate].
            nodo,this.caminoTotal[locate].no do.GetComponent<NodoV2>()
            .NodeCollection[i]));
        this.caminoTotal.Add(extra);
    }
}

public float MedirHeuristica(GameObject ini, GameObject fin){
    float res = 0;

```

```

        res += Vector3.Distance(ini.transform.position,
fin.transform.
position);
        res += Vector3.Distance(fin.transform.position, this.final.
transform.position);
        return res;
    }
}
[System.Serializable]
public class NodoEspecial{
    public GameObject nodo;
    public int paso;
    public bool actual;
    public float heuris;
    public NodoEspecial(GameObject nodo, int paso, bool actual,
float heuris){
        this.nodo = nodo;
        this.paso = paso;
        this.actual = actual;
        this.heuris = heuris;
    }
}
}

```

4.3.2.3. Construcción de la Base de Datos

Script de la Base de Datos:

```

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

CREATE TABLE `arista` (
  `cod_arista` int(11) NOT NULL,
  `cod_nodo_1` int(11) NOT NULL,
  `cod_nodo_2` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `destino` (
  `cod_destino` int(11) NOT NULL,
  `nombre` varchar(30) NOT NULL,
  `descripcion` text NOT NULL,
  `cod_nodo` int(11) NOT NULL,
  `cod_usuario` int(11) NOT NULL,
  `imagen` blob NOT NULL,
  `estado` tinyint(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `nodo` (
  `cod_nodo` int(11) NOT NULL,
  `nombre` varchar(30) NOT NULL,
  `longitud` double NOT NULL,
  `latitud` double NOT NULL,
  `cod_usuario` int(11) NOT NULL,
  `estado` tinyint(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `usuario` (
  `Cod_usuario` int(11) NOT NULL,
  `Nombre` varchar(30) NOT NULL,

```

```

`Ap_paterno` varchar(30) NOT NULL,
`Ap_materno` varchar(30) NOT NULL,
`Correo` varchar(30) NOT NULL,
`Usuario` varchar(30) NOT NULL,
`Contraseña` varchar(30) NOT NULL,
`Tipo` tinyint(1) NOT NULL,
`Estado` tinyint(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `arista`
  ADD PRIMARY KEY (`cod_arista`),
  ADD KEY `cod_nodo_1` (`cod_nodo_1`),
  ADD KEY `cod_nodo_2` (`cod_nodo_2`);

ALTER TABLE `destino`
  ADD PRIMARY KEY (`cod_destino`),
  ADD KEY `cod_nodo` (`cod_nodo`),
  ADD KEY `cod_usuario` (`cod_usuario`);

ALTER TABLE `nodo`
  ADD PRIMARY KEY (`cod_nodo`),
  ADD KEY `cod_usuario` (`cod_usuario`);

ALTER TABLE `usuario`
  ADD PRIMARY KEY (`Cod_usuario`);

ALTER TABLE `arista`
  MODIFY `cod_arista` int(11) NOT NULL AUTO_INCREMENT;

ALTER TABLE `destino`
  MODIFY `cod_destino` int(11) NOT NULL AUTO_INCREMENT;

ALTER TABLE `nodo`
  MODIFY `cod_nodo` int(11) NOT NULL AUTO_INCREMENT;
ALTER TABLE `usuario`
  MODIFY `Cod_usuario` int(11) NOT NULL AUTO_INCREMENT;

ALTER TABLE `arista`
  ADD CONSTRAINT `arista_ibfk_1` FOREIGN KEY (`cod_nodo_1`)
REFERENCES `nodo` (`cod_nodo`),
  ADD CONSTRAINT `arista_ibfk_2` FOREIGN KEY (`cod_nodo_2`)
REFERENCES `nodo` (`cod_nodo`);

ALTER TABLE `destino`
  ADD CONSTRAINT `destino_ibfk_1` FOREIGN KEY (`cod_nodo`)
REFERENCES `nodo` (`cod_nodo`),
  ADD CONSTRAINT `destino_ibfk_2` FOREIGN KEY (`cod_usuario`)
REFERENCES `usuario` (`Cod_usuario`);

ALTER TABLE `nodo`
  ADD CONSTRAINT `nodo_ibfk_1` FOREIGN KEY (`cod_usuario`)
REFERENCES `usuario` (`Cod_usuario`);
COMMIT;

```

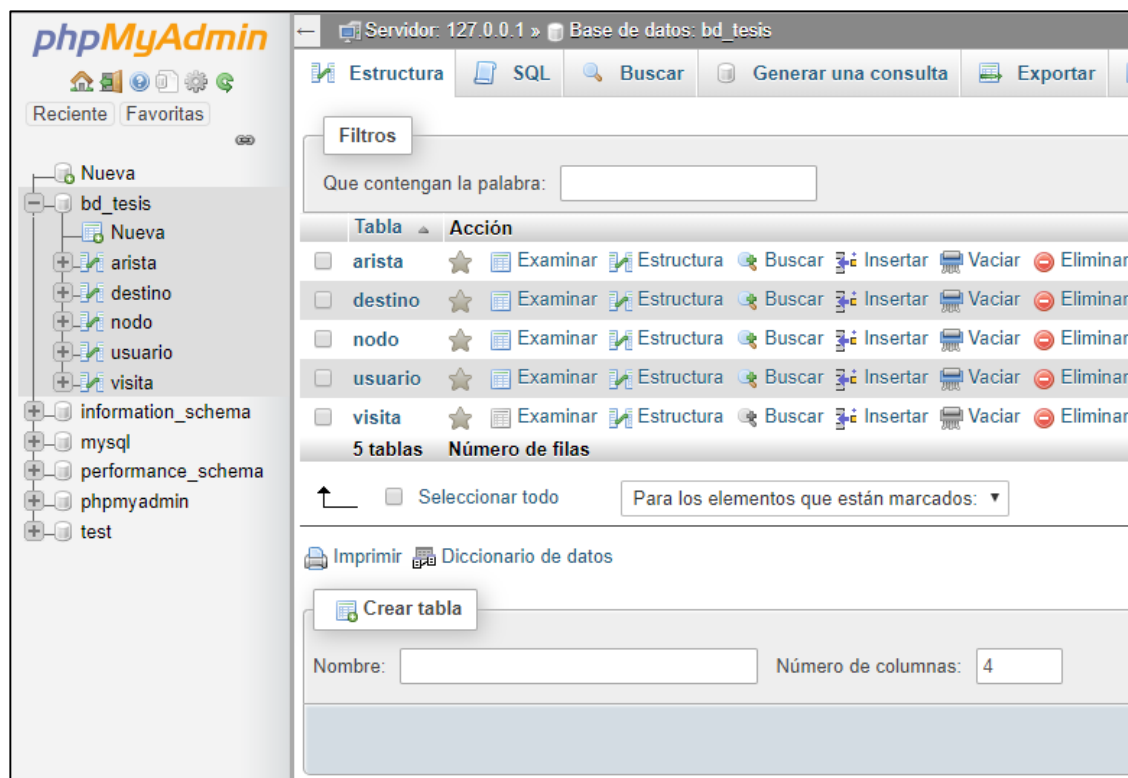


Figura 47. Construcción de la Base de Datos

Fuente: Elaboración Propia

Antes de finalizar con la base de datos, debemos de generar un script SQL para insertar como registro inicial a un usuario administrador que se encargará de realizar las pruebas del sistema:

```
INSERT INTO `usuario` (`cod_usuario`, `nombre`, `ap_pat`, `ap_mat`,
`correo`, `usuario`, `contraseña`, `tipo`, `estado`) VALUES (NULL,
'Administrador', 'Administrador', 'Administrador',
'Ejemplo@gmail.com', 'Admin', 'Admin', 'A', '1');
```

4.3.2.4. Construcción de la Aplicación Web

La aplicación web, se construirá en Sublime Text con los lenguajes PHP, HTML5 y CSS3 (Ver Anexo 6). Su estructura constara de 4 paquetes tal y como la muestra la Figura 48.

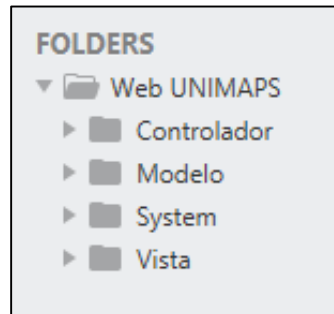


Figura 48. Estructura de la Aplicación Web
Fuente: Elaboración Propia

Cada paquete contendrá los siguientes componentes:

- Vista: 11 Script HTML5
- Controlador: 2 Script PHP
- Modelo: 10 Script PHP
- System: 1 Script PHP, 2 Script CSS3, 2 Script JavaScript y 1 imágenes jpg

4.3.2.5. Construcción de la Interfaz de Programación de Aplicaciones

La API, se construirá en Sublime Text con el lenguaje de programación PHP (Ver Anexo 7). Constará de cuatro scripts almacenados en un único paquete tal y como muestra la Figura 49.

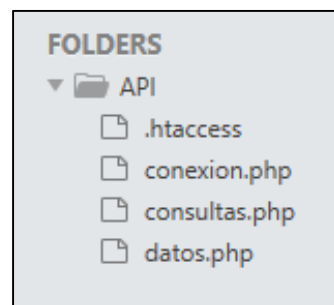


Figura 49. Estructura de la Interfaz de Programación de Aplicaciones
Fuente: Elaboración Propia

4.3.2.6. Construcción de la Aplicación Móvil

La aplicación móvil, se construirá en Unity 3D con el lenguaje de programación C# y objetos 3D y 2D (Ver Anexo 8). Su estructura constara de 4 paquetes tal y como la muestra la Figura 50.



Figura 50. Estructura de la Aplicación Móvil
Fuente: Elaboración Propia

Cada paquete contendrá los siguientes componentes:

- Datos: 2 Scripts C#
- Lógica: 15 Scripts C#
- Materiales: 44 Objeto 3D y 8 Objeto 2D
- Scenes: Interfaz gráfica

Como precondition final para exportar la aplicación a la plataforma

Android, Unity 3d requerirá importar los siguientes componentes:

- SDK de Android
- NDK de Android
- JDK de Java

4.4. TRANSICIÓN

4.4.1. Pruebas del Sistema

4.4.1.1. Tipo de Prueba

Pruebas de componentes:

También se denominan pruebas unitarias o de desarrollo, consiste en probar la funcionalidad de cada componente individual de manera separada y aislada. Se trata de las pruebas que se realizan sobre el software implementado y son las pruebas de cada uno de los componentes básicos del software respecto de su programación. Así podrían ser pruebas de módulo, de clases, de unidad, etc... Y su finalidad es asegurar la funcionalidad del componente dado intentando localizar errores en la elaboración o ausencia de funciones especificadas en los requisitos. (Mera, 2016, p.170)

4.4.1.2. Técnica Utilizada

Pruebas de la caja negra:

En este tipo de pruebas la estructura del programa es irrelevante o desconocida, o dicho de otra forma, es transparente para el probador ya que la importancia de este tipo de pruebas reside en la funcionalidad como tal. Es por ello, que en muchas ocasiones se las conoce como pruebas funcionales. Las pruebas se reducen en su totalidad a las especificaciones de los requisitos, se trata de pruebas en las que se valida el comportamiento de entrada y de salida de un cierto elemento. (Mera, 2016, p.171)

4.4.1.3. Catálogo de pruebas

A continuación, se muestra en la Tabla 26 las pruebas realizadas al sistema:

Tabla 26.
Catálogo de Pruebas

Código	Modulo	Caso de Uso	Objetivo de la prueba
PU-001	Usuario	Cargar Datos	Verificar en el sistema el retorno de resultados al cargar los datos en la aplicación móvil.
PU-002	Usuario	Consultar Origen	Verificar en el sistema el retorno de resultados al seleccionar opción Consultar Origen.
PU-003	Usuario	Consultar Destino	Verificar en el sistema el retorno de resultados al seleccionar opción Consultar Destino.
PU-004	Usuario	Buscar Ruta	Verificar en el sistema el retorno de resultados al seleccionar la opción buscar ruta.
PU-005	Administración	Iniciar Sesión	Verificar en el sistema el retorno de resultados al ingresar una contraseña incorrecta o vacía.
PU-006	Administración	Iniciar Sesión	Verificar en el sistema el retorno de resultados al ingresar un Id incorrecto o vacío.
PU-007	Administración	Gestionar Usuario	Verificar que se puedan agregar usuarios al sistema.
PU-008	Administración	Gestionar Usuario	Verificar que no se pueda agregar un usuario con un login ya registrado en el sistema.
PU-009	Administración	Gestionar Usuario	Verificar que se pueda modificar un usuario del sistema.

Fuente: Elaboración propia

Continuación de la Tabla 26, Catálogo de Pruebas

PU-010	Administración	Gestionar Usuario	Verificar que no se pueda agregar en el sistema un usuario si falta llenar un campo obligatorio.
PU-011	Administración	Gestionar Usuario	Verificar que se pueda eliminar un usuario del sistema.
PU-012	Administración	Gestionar Nodo	Verificar que se puedan agregar Nodos al sistema
PU-013	Administración	Gestionar Nodo	Verificar que se pueda modificar un nodo del sistema
PU-014	Administración	Gestionar Nodo	Verificar que no se pueda agregar un nodo con un nombre ya registrado en el sistema.
PU-015	Administración	Gestionar Nodo	Verificar que no se pueda agregar en el sistema un nodo si falta llenar un campo obligatorio
PU-016	Administración	Gestionar Nodo	Verificar que se pueda eliminar un nodo del sistema.
PU-017	Administración	Gestionar Destino	Verificar que se puedan agregar destinos al sistema
PU-018	Administración	Gestionar Destino	Verificar que se pueda modificar un destino del sistema
PU-019	Administración	Gestionar Destino	Verificar que no se pueda agregar un destino con un nombre ya registrado en el sistema.
PU-020	Administración	Gestionar Destino	Verificar que no se pueda agregar en el sistema un destino si falta llenar un campo obligatorio
PU-021	Administración	Gestionar Destino	Verificar que se pueda eliminar un destino del sistema.
PU-022	Administración	Gestionar Arista	Verificar que se puedan agregar aristas al sistema

Fuente: Elaboración propia

Continuacion de la Tabla 26, Catálogo de Pruebas

PU-023	Administración	Gestionar Arista	Verificar que se pueda modificar una arista del sistema
PU-024	Administración	Gestionar Arista	Verificar que no se pueda agregar en el sistema una arista si falta llenar un campo obligatorio
PU-025	Administración	Gestionar Arista	Verificar que se pueda eliminar una arista del sistema
PU-026	Administración	Generar Grafo	Verificar en el sistema el retorno de resultados al generar grafo
PU-027	Administración	Generar Reporte Visitantes	Verificar que se pueda mostrar la cantidad de visitantes por día.

Fuente: Elaboración propia

Las pruebas unitarias detalladas en el catálogo de pruebas fueron evaluadas en función a las entradas realizadas por el tester y las salidas obtenidas del sistema. (Ver Anexo 5). El resultado obtenido fue de un 100% de éxito en función a las pruebas superadas.

4.4.2. Pruebas con los usuarios

Se realizaron las pruebas con la aplicación móvil que se ha propuesto para poder determinar el cumplimiento de los objetivos, para lo cual se utilizó un muestreo no probabilístico de tipo convencional, estableciéndose una cuota de 20 personas por conveniencia de los autores que a su vez utilizaron la aplicación móvil para realizar un cuestionario, el cual consta de 7 preguntas. (Ver Anexo 8), obteniendo los siguientes resultados:

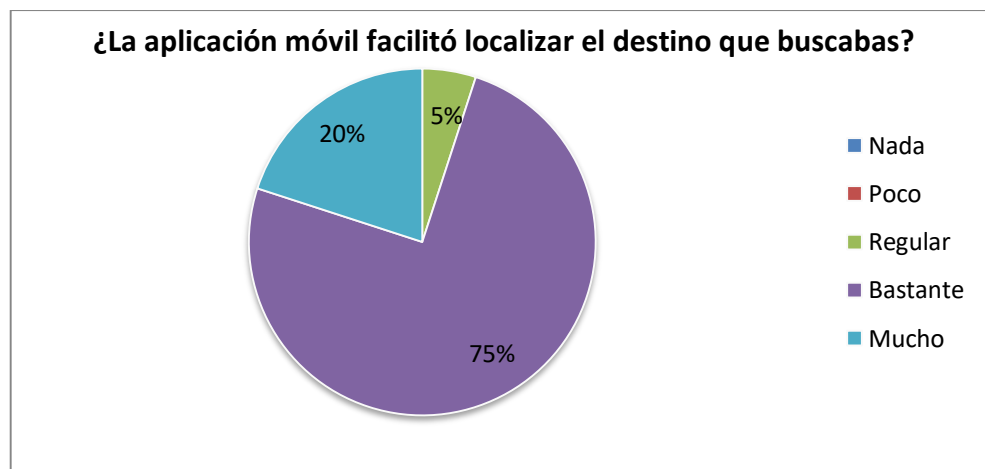


Figura 51: Visualización Estadística de la Encuesta N° 2 Pregunta 1

Fuente: Elaboración Propia

En el gráfico de la figura 49, respondiendo a la pregunta: “¿La aplicación móvil facilitó localizar el destino que buscabas?”, el 75% de los usuarios indicaron que bastante y un 20%, mucho. Demostrando que la aplicación móvil facilitó la localización de los destinos a la mayoría de usuarios.

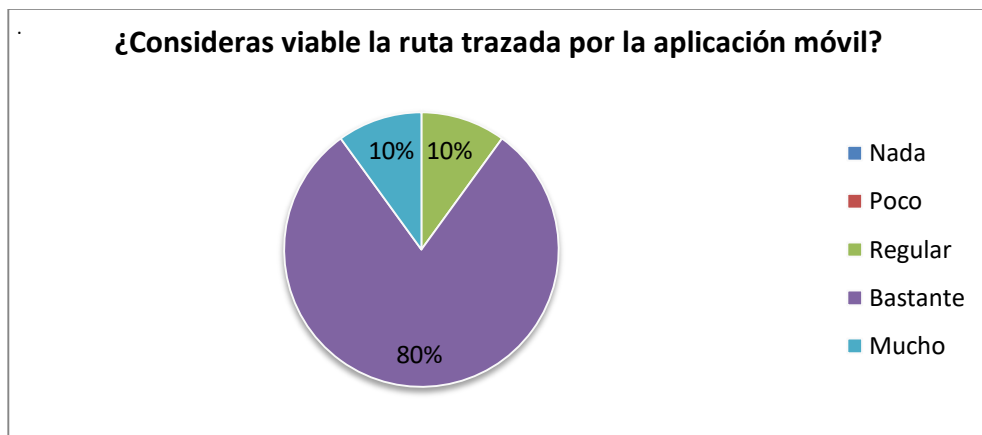


Figura 52: Visualización Estadística de la Encuesta N° 2 Pregunta 2
Fuente: Elaboración Propia

En el gráfico de la figura 50, respondiendo a la pregunta: “¿Consideras viable la ruta trazada por la aplicación móvil?”, el 80% de los usuarios indicaron que bastante y un 10%, mucho. Demostrando que las rutas trazadas por la aplicación móvil fueron satisfactoria para la mayoría de usuarios.

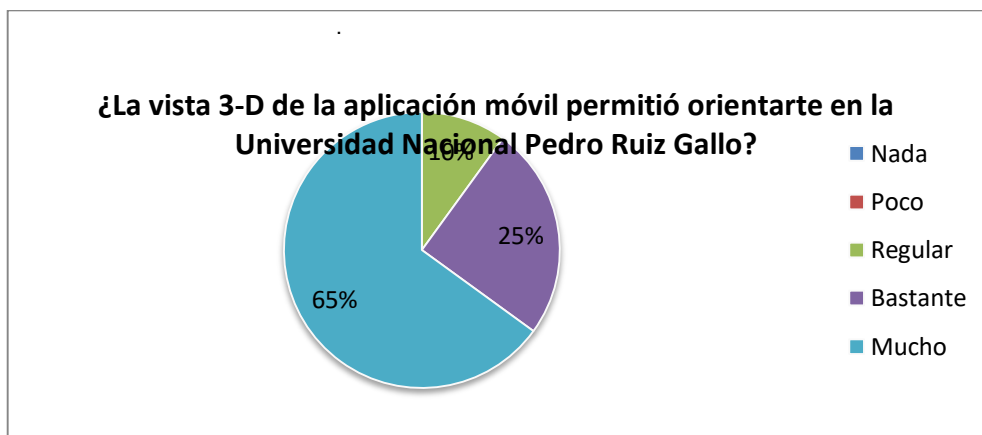


Figura 53: Visualización Estadística de la Encuesta N° 2 Pregunta 3
Fuente: Elaboración Propia

En el gráfico de la figura 51, respondiendo a la pregunta: “¿La vista 3-D de la aplicación móvil permitió orientarte en la Universidad Nacional Pedro Ruiz Gallo?”, el 25% de los usuarios indicaron que bastante y un 65%, mucho. Demostrando que el diseño 3-D de la aplicación móvil permitió reconocer las rutas con mayor facilidad a la mayoría de los usuarios.

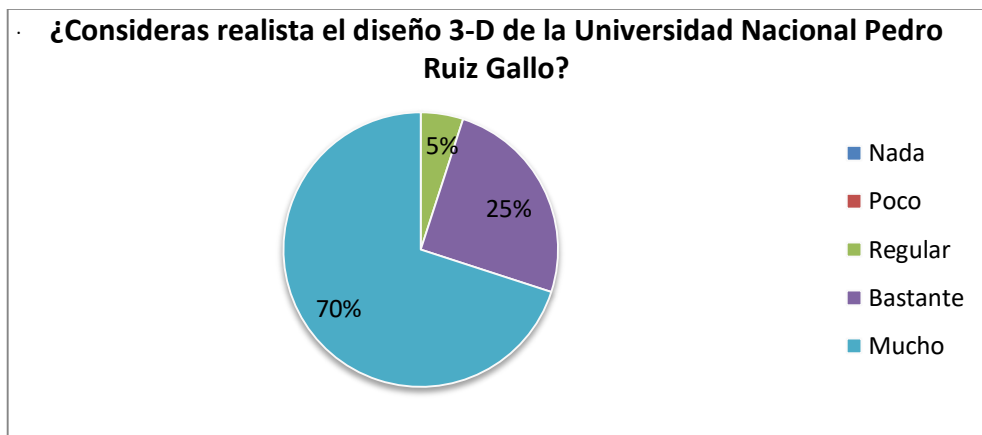


Figura 54: Visualización Estadística de la Encuesta N° 2 Pregunta 4
Fuente: Elaboración Propia

En el gráfico de la figura 52, respondiendo a la pregunta: “¿Consideras realista el diseño 3-D de la Universidad Nacional Pedro Ruiz Gallo?”, el 25% de los usuarios indicaron que bastante y el 70%, mucho. Demostrando que el diseño grafico 3-D de la aplicación móvil fue agradable para la mayoría de usuarios.

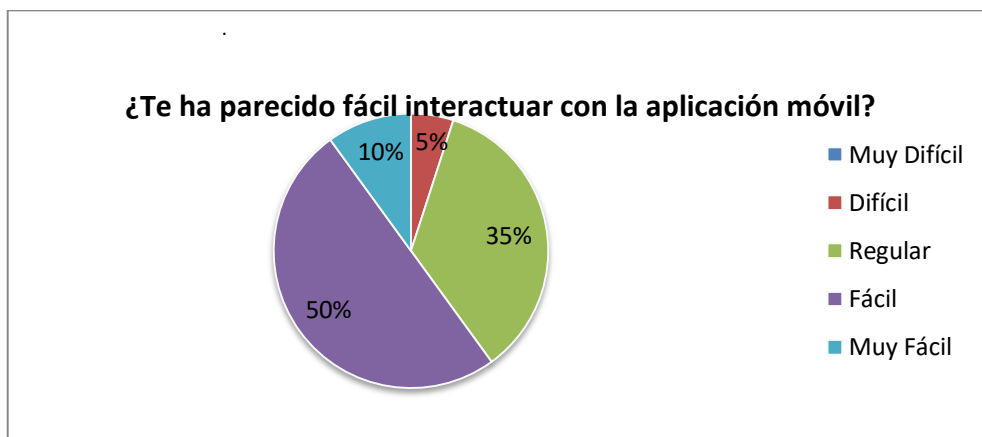


Figura 55: Visualización Estadística de la Encuesta N° 2 Pregunta 5
Fuente: Elaboración Propia

En el gráfico de la figura 53, respondiendo a la pregunta: “¿Te ha parecido fácil interactuar con la aplicación móvil?”, el 50% de los usuarios indicaron que fácil y un 10%, muy fácil. Demostrando que la usabilidad de la aplicación móvil fue agradable para la mayoría de usuario.

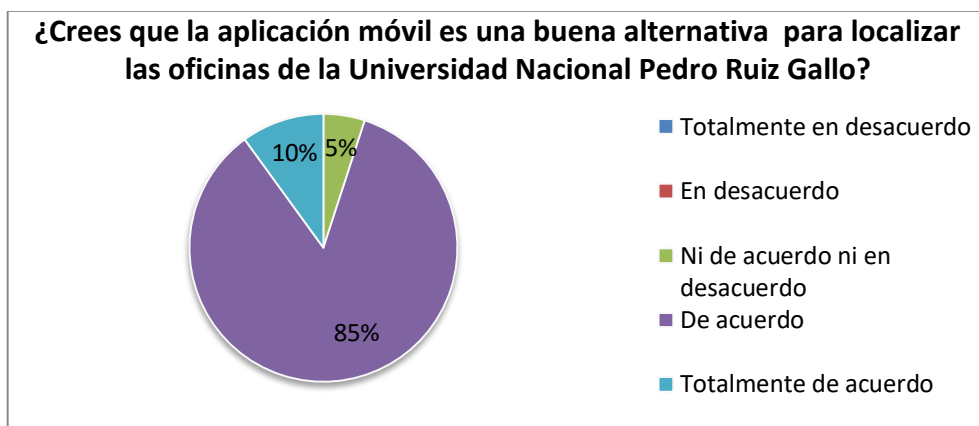


Figura 56: Visualización Estadística de la Encuesta N° 2 Pregunta 6
Fuente: Elaboración Propia

En el gráfico de la figura 54, respondiendo a la pregunta: “¿Crees que la aplicación móvil es una buena alternativa para localizar las oficinas de la Universidad Nacional Pedro Ruiz Gallo?”, el 80% de los usuarios estuvieron de acuerdo y un 10%, Totalmente de acuerdo. Demostrando que la aplicación móvil es considerada por la mayoría de usuarios como una buena alternativa para la búsqueda de rutas en la universidad.

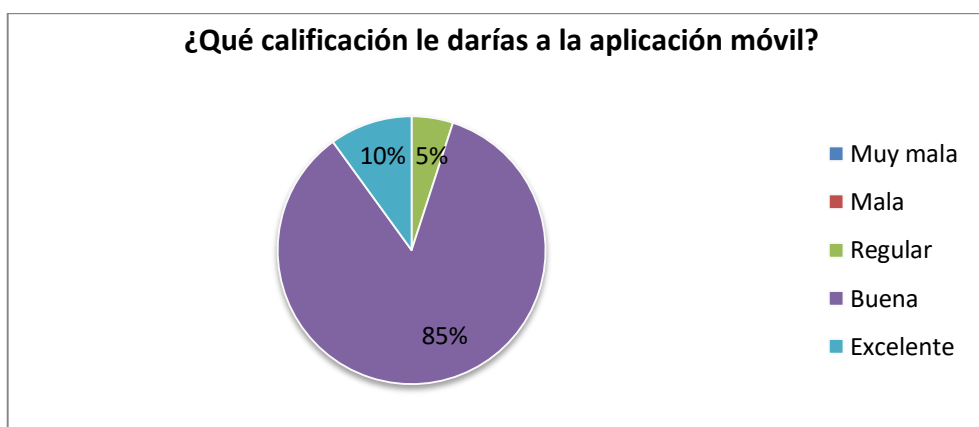


Figura 57: Visualización Estadística de la Encuesta N° 2 Pregunta 7
Fuente: Elaboración Propia

En el gráfico de la figura 55, respondiendo a la pregunta: “¿Qué calificación le darías a la aplicación móvil?”, el 80% de los usuarios indicaron que buena y un 10%, Excelente. Demostrando que la aplicación móvil es considerada agradable por la mayoría de usuarios.

CONCLUSIONES

1. El desarrollo del sistema de búsqueda heurística permitió localizar las principales oficinas de la Universidad Nacional Pedro Ruiz Gallo con mayor facilidad gracias a las rutas generadas por el algoritmo A star.
2. La metodología AUP ofreció un marco de trabajo estructurado en la elaboración del sistema de búsqueda heurística gracias a las fases de iniciación, elaboración, construcción y transición.
3. La herramienta Blender facilitó el modelado tridimensional de la estructura de la Universidad Nacional Pedro Ruiz Gallo y la aplicación de texturas para darle más realismo al diseño.
4. El algoritmo de búsqueda seleccionado para la presente investigación fue A*, porque se desenvuelve mejor en un grafo bidireccional sin importar la distribución de los nodos o el coste de las aristas.
5. Se estableció 435 nodos para las vías transitables, entre los cuales se consideró veredas, aceras y la señalización de cruces peatonales de la Universidad Nacional Pedro Ruiz Gallo, que permitirá realizar las búsquedas en el sistema de búsqueda heurística.
6. Se desarrolló una base de datos en MySQL que contiene cuatro tablas: arista, nodo, destino y usuario para la administración del grafo y de los usuarios del sistema de búsqueda heurística, que será necesarios cuando se construya un nuevo local con oficinas administrativas o cuando se cambie de ubicación las oficinas existentes.
7. Las 27 pruebas de los módulos del sistema: realizados en el catálogo de pruebas dieron como resultado que funcionaban correctamente cada uno de los componentes del software.

RECOMENDACIONES

1. El sistema de búsqueda heurística para la localización de las principales oficinas de la Universidad Nacional Pedro Ruíz Gallo fue desarrollado para ser utilizado exclusivamente en la plataforma Android, sin embargo el proyecto puede ser adaptado a otras plataformas puesto que el motor Unity 3D presenta la opción de compilación para una variedad de apartados de sistemas operativos.
2. Al desarrollar la metodología AUP se obtiene una mejor interpretación cuando se combina con los diagramas de UML.
3. La herramienta Blender tiene la opción de crear una plantilla a partir del mapeado de un objeto, es altamente recomendable usar esta plantilla en un editor de imágenes facilitando así la creación de las texturas.
4. Se recomienda utilizar el algoritmo A Star cuando el entorno permita obtener una calidad admisible en la heurística.
5. Se recomienda ubicar los nodos en vértices y en puntos adyacentes a las veredas de la universidad para que el grafo represente con mayor exactitud las vías transitables por las personas.
6. Se recomienda generar un script SQL para insertar como registro inicial a un usuario administrador que nos permita loguearnos en la página web y realizar el registro de todos datos.
7. Al programar se recomienda evitar escribir funciones y procedimientos demasiado largos, los cuales dificultan la realización pruebas.

BIBLIOGRAFIA

- 3D CAD Portal. (2018). *AUTOCAD*. Obtenido de <http://www.3dcadportal.com/3d-software/cad/>
- Arias Rojas, J. S. (2010). *Aplicación de un modelo de optimización en la planeación de rutas de los buses escolares del colegio Liceo de Cervantes Norte*. Obtenido de Tesis para la obtención del título de Ingeniero Industrial:
<https://repository.javeriana.edu.co/bitstream/handle/10554/7367/tesis403.pdf;sequence=1>
- Carrión Castagnola, P. J. (2016). *Visualización de puntos de interés en un campus universitario usando realidad aumentada*. Obtenido de Tesis para obtener el Título de Ingeniero Informático:
<http://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/6848>
- Collell Puig, J. (2015). *CSS3 y Javascript avanzado*. Obtenido de Universidad Abierta de Cataluña:
[https://www.exabyteinformatica.com/uoc/Informatica/Tecnologias_y_herramientas_para_el_desarrollo_web/Tecnologias_y_herramientas_para_el_desarrollo_web_\(Modulo_1\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Tecnologias_y_herramientas_para_el_desarrollo_web/Tecnologias_y_herramientas_para_el_desarrollo_web_(Modulo_1).pdf)
- García Fernandez, J., Fernández Gavira, J., Galvez Ruiz, P., Sanchez Oliver, A., & Grimaldi Puyana, M. (2017). *La Geolocalización*. Obtenido de La Geolocalización como Herramienta para Conocer a Los Consumidores:
https://www.researchgate.net/publication/319654852_La_Geolocalizacion_como_Herramienta_para_Conocer_a_los_Consumidores_El_Caso_de_los_Centros_de_Fitness
- Gilfillan, I. (2003). *MySQL*. Obtenido de La Biblia de MySQL:
<http://didepa.uaemex.mx/clases/Manuales/MySql/MySqlLa%20biblia%20de%20mysql.pdf>
- GNU Image Manipulation Program. (2005). *Manual de GIMP*. Obtenido de
<http://dis.um.es/~jfernand/0506/smig/gimp.pdf>
- Gómez Fuentes, M. (2011). *Análisis de Requerimientos*. Obtenido de
<http://repositorio.uncp.edu.pe/bitstream/handle/UNCP/1153/DISE%C3%91O%20DE%20UN%20SISTEMA%20BASADO%20EN%20TECNOLOG%C3%8DA%20WEB.pdf?sequence=1&isAllowed=y>
- IEEE Std 830. (1998). *IEEE Recommended Practice for Software Requirements Specifications*. Obtenido de
<http://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf>
- Li Fernández, M., & Fonseca Romero, S. (2013). *Desarrollo de una herramienta software para determinar la ruta de acarreo de mineral mas corta*. Obtenido de Tesis para

- optener el Título de Ingeniero Informático:
<http://dspace.unitru.edu.pe/bitstream/handle/UNITRU/8556/LI%20FERN%C3%81N%20DEZ%2c%20Mar%C3%ADA%20H.%20-%20FONSECA%20ROMERO%2c%20Sandro%20H.pdf?sequence=1&isAllowed=y>
- Luttecke, C. (2014). *Powered By Unity*. Obtenido de <https://deideaaapp.org/sabes-que-es-unity-descubrelo-aqui/>
- Marchena Pejerrey, D. J. (2015). *Desarrollo de un sistema para la optimización de rutas de trabajo utilizando el algoritmo de Dijkstra y diagramas de Voronoi*. Obtenido de Tesis para obtener el Título de Ingeniero de Sistemas:
<http://repositorio.uss.edu.pe/handle/uss/157>
- Martínez E, F., & Sainz, G. (2010). *Modelos y algoritmos de optimización combinatoria para planificación de rutas en regatas de barcos de vela*. Obtenido de Tesis para obtener el Título de Ingeniero de Sistemas: <https://www.dc.uba.ar/academica/tesis-de-licenciatura/2010/martinez-sainz-trapaga>
- Martínez Echevarría, Á. (2002). *Manual práctico de HTML*. Obtenido de Universidad Politécnica de Madrid:
<http://bioinf.ibun.unal.edu.co/servicios/electiva/manhtml/HTML.pdf>
- Meneses Pilco, S. A. (2014). *Diseño de un algoritmo genético para la optimización de distancias en ambientes tridimensionales*. Obtenido de Tesis para optar el Título de Ingeniero Informático:
http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/5723/MENESES_SEBASTIAN_ALGORITMO_GENETICO_DISTANCIAS_AMBIENTES_TRIDIMENSIONALES.pdf?sequence=1&isAllowed=y
- Mera Paz, J. A. (2016). *Análisis del proceso de pruebas*. Obtenido de [file:///C:/Users/USUARIO/Downloads/1482-Article%20Text-3592-2-10-20170131%20\(3\).pdf](file:///C:/Users/USUARIO/Downloads/1482-Article%20Text-3592-2-10-20170131%20(3).pdf)
- Microsoft. (2015). *Guía de C#*. Obtenido de Microsoft: <https://docs.microsoft.com/es-es/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>
- Núñez Mori, J. (2010). *Usabilidad en Metodologías Ágiles*. Obtenido de Universidad Politécnica de Madrid: https://www.fi.upm.es/catedra-ibmrational/sites/www.fi.upm.es/catedra-ibmrational/files/Tesis_MarcoAgilTrabajo.pdf

- Project Management Institute. (2013). *Guía de los Fundamentos para la Dirección de Proyectos* (Quinta ed.). Estados Unidos de América: National Information Standards Organization.
- Sánchez Osorio, A. F., & Parra Rativa, L. A. (2017). *Manual del Programador*. Obtenido de Universidad Distrital Francisco José de Caldas:
<http://repository.udistrital.edu.co/bitstream/11349/5873/3/ParraRativaLuisAlfredo2017AnexoAnexo%202.pdf>
- Reyes Esquén, J. O. (2018). *Desarrollo de un planificador de rutas para recojo de desechos sólidos utilizando algoritmo de Bellman Ford*. Obtenido de Tesis para obtener el Título de Ingeniero de Sistemas:
<http://repositorio.uss.edu.pe/bitstream/handle/uss/5779/Reyes%20Esqu%c3%a9n%20Jeremy%20Octavio.pdf?sequence=1&isAllowed=y>
- Ros Bernabeu, M. E., & Giménez Rodríguez, T. (2010). *Sistema de Posicionamiento Global*. Obtenido de https://webs.um.es/bussons/GPSresumen_TamaraElena.pdf
- Russell, S. J., & Norvig, P. (2008). *Inteligencia Artificial un Enfoque Moderno* (Segunda ed.). México: Pearson Prentice Hall.
- The PHP Group. (2019). *PHP*. Obtenido de Manual de PHP:
http://php.net/get/php_manual_es.html.gz/from/a/mirror
- Trammell, K. (2012). *¿Qué hace Blender?* Obtenido de Instituto Nacional de Tecnología Educación y de Formación del Profesional:
http://www.ite.educacion.es/formacion/materiales/181/cd/m1/qu_hace_blender.html
- Tutorials Point. (2015). *Sublime Text*. Obtenido de
https://www.tutorialspoint.com/sublime_text/sublime_text_tutorial.pdf
- Universidad de Pamplona. (2012). *Teoría de grafos*. Obtenido de
http://www.unipamplona.edu.co/unipamplona/portaIG/home_23/recursos/general/11072012/grafos3.pdf
- Universidad Don Bosco. (2017). *Algoritmos para la ruta más corta en un grafo*. Obtenido de <http://www.udb.edu.sv/udb/archivo/guia/informatica-ingenieria/programacion-iv/2017/ii/guia-10.pdf>

ANEXOS

Anexo 1: Encuesta para medir el conocimiento de la localización de las principales Oficinas de la Universidad Nacional Pedro Ruiz Gallo

Nombre:..... Código:.....



1. Al realizar un trámite académico, ¿alguna vez has tenido dificultad en encontrar alguna oficina de la universidad?
 - a. Nunca
 - b. Escasamente
 - c. Ocasionalmente
 - d. frecuentemente
 - e. Siempre
2. ¿Usted conoce la ubicación de la Facultad de Ciencias Físicas y Matemáticas?
 - a. Si
 - b. No
3. ¿Usted conoce la ubicación de la Facultad de Derecho y Ciencias Políticas?
 - a. Si
 - b. No
4. ¿Usted conoce la ubicación de la Facultad de Agronomía?
 - a. Si
 - b. No
5. ¿Usted conoce la ubicación de la Facultad de Ciencias Biológicas?
 - a. Si
 - b. No

6. ¿Usted conoce la ubicación de la Facultad de Enfermería?
- a. Si
 - b. No
7. ¿Usted conoce la ubicación de la Facultad de Ciencias en Economía, Administración y Contabilidad?
- a. Si
 - b. No
8. ¿Usted conoce la ubicación de la Facultad de Ciencias Histórico Social y Educación?
- a. Si
 - b. No
9. ¿Usted conoce la ubicación de la Facultad de Ingeniería Agrícola?
- a. Si
 - b. No
10. ¿Usted conoce la ubicación de la Facultad de Ingeniería en Zootecnia?
- a. Si
 - b. No
11. ¿Usted conoce la ubicación de la Facultad de Medicina Humana?
- a. Si
 - b. No
12. ¿Usted conoce la ubicación de la Facultad de Ingeniería Mecánica y Eléctrica?
- a. Si
 - b. No
13. ¿Usted conoce la ubicación de la Facultad de Medicina Veterinaria?
- a. Si
 - b. No
14. ¿Usted conoce la ubicación de la Facultad de Ingeniería Civil, Sistemas y Arquitectura?
- a. Si
 - b. No
15. ¿Usted conoce la ubicación de la Facultad de Química e Industrias Alimentarias?

- a. Si
 - b. No
16. ¿Usted conoce la ubicación de la Oficina de Rectorado?
- a. Si
 - b. No
17. ¿Usted conoce la ubicación de la Oficina de Vicerrectorado?
- a. Si
 - b. No
18. ¿Usted conoce la ubicación de la Oficina Central de Asuntos Académicos?
- a. Si
 - b. No
19. ¿Usted conoce la ubicación de la Oficina Central de Investigación?
- a. Si
 - b. No
20. ¿Usted conoce la ubicación de la Oficina Central de Proyección Social?
- a. Si
 - b. No
21. ¿Usted conoce la ubicación de la Oficina Central de Biblioteca?
- a. Si
 - b. No
22. ¿Usted conoce la ubicación de la Oficina Central de Admisión?
- a. Si
 - b. No
23. ¿Usted conoce la ubicación de la oficina central de sistemas informáticos?
- a. Si
 - b. No
24. ¿Usted conoce la ubicación de la Oficina de Grados y Títulos?
- a. Si
 - b. No

Tabla 27.

Encuesta N° 1, Respuestas de las personas encuestadas

Encuestados		Preguntas																							
Código		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
171769H	Escasamente	SI	SI	NO	SI	SI	NO	NO	SI	NO	NO	NO	NO	SI	SI	SI	SI	NO	NO	NO	SI	NO	NO	NO	NO
171770F	Ocasionalmente	SI	SI	SI	SI	SI	SI	NO	SI	NO	NO	SI	NO	SI	SI	SI	SI	NO	NO	NO	SI	NO	SI	NO	NO
170533K	Frecuentemente	SI	NO	NO	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	SI	NO	SI	NO
170528G	Nunca	SI	SI	SI	SI	NO	NO	SI	SI	NO	SI	NO	SI	NO	SI	SI	NO	SI	NO	NO	SI	NO	NO	NO	NO
171771B	Frecuentemente	SI	SI	SI	SI	SI	SI	SI	SI	NO	NO	SI	NO	SI	SI	SI	SI	NO	SI	NO	SI	NO	SI	NO	NO
171775H	Escasamente	SI	SI	NO	SI	NO	SI	NO	SI	NO	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	NO	NO	NO	NO
170463B	Frecuentemente	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO	NO	SI	SI	SI	SI	NO	SI	NO	NO	SI	NO	NO	NO	NO
170444H	Siempre	SI	SI	SI	NO	SI	NO	SI	NO	NO	SI	SI	NO	SI	NO	NO	NO	NO	NO	NO	SI	NO	NO	NO	NO
170431C	Ocasionalmente	SI	SI	NO	SI	SI	SI	SI	SI	NO	NO	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	NO	SI	NO	NO
171778G	Frecuentemente	SI	SI	SI	SI	SI	NO	NO	SI	NO	NO	SI	SI	SI	NO	SI	NO	SI	NO	NO	SI	NO	NO	NO	NO
162380D	Frecuentemente	SI	NO	SI	SI	SI	SI	SI	SI	NO	SI	SI	NO	SI	SI	SI	SI	NO	NO	NO	SI	SI	SI	NO	NO
170502H	Escasamente	SI	SI	NO	SI	SI	SI	SI	NO	SI	NO	SI	SI	SI	SI	NO	SI	NO	NO	NO	SI	NO	NO	NO	NO
160250F	Escasamente	SI	SI	SI	SI	NO	SI	SI	SI	SI	NO	SI	SI	SI	SI	SI	SI	SI	NO	NO	SI	NO	NO	NO	NO
170496H	Ocasionalmente	SI	SI	NO	SI	SI	SI	NO	SI	NO	NO	SI	SI	NO	NO	SI	SI	NO	NO	NO	SI	NO	SI	SI	NO
171780A	Frecuentemente	SI	SI	SI	SI	SI	SI	SI	SI	NO	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	NO	SI	NO	NO
179055D	Frecuentemente	SI	SI	SI	SI	SI	NO	SI	SI	SI	NO	SI	SI	SI	SI	SI	SI	SI	NO	NO	SI	SI	NO	NO	NO
171781H	Escasamente	SI	SI	NO	SI	SI	SI	SI	SI	NO	NO	NO	SI	SI	SI	SI	SI	NO	NO	NO	SI	NO	SI	NO	NO
165702B	Escasamente	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO	SI	SI	SI	SI	SI	NO	NO	NO	NO	SI	NO	NO	NO	NO
162377C	Frecuentemente	SI	SI	NO	SI	SI	NO	SI	SI	NO	NO	NO	SI	SI	SI	SI	SI	NO	NO	NO	SI	NO	NO	NO	NO
175577F	Ocasionalmente	SI	SI	NO	SI	SI	SI	SI	NO	NO	SI	SI	NO	SI	NO	SI	SI	SI	NO	NO	SI	NO	SI	NO	NO

Fuente: Elaboración propia

Continuación de la Tabla 27, Respuestas de las personas encuestadas

162384J	Frecuentemente	SI	SI	SI	SI	SI	SI	NO	SI	SI	NO	SI	SI	NO	SI	SI	SI	NO	NO	NO	SI	NO	NO	NO
170735B	Escasamente	SI	NO	SI	SI	SI	SI	SI	SI	NO	NO	SI	SI	SI	SI	SI	SI	SI	NO	NO	SI	NO	NO	NO
162437F	Frecuentemente	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	NO	NO	NO
171783K	Ocasionalmente	SI	SI	SI	SI	NO	NO	SI	SI	NO	NO	NO	SI	SI	SI	SI	SI	NO	SI	NO	SI	NO	SI	NO
165072I	Escasamente	SI	SI	SI	SI	SI	SI	SI	SI	NO	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	NO	SI	NO
171786J	Nunca	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI
162398K	Escasamente	SI	SI	NO	SI	SI	SI	SI	NO	SI	SI	SI	NO	SI	SI	SI	SI	SI	NO	NO	SI	NO	NO	SI
170678I	Frecuentemente	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	NO	NO	NO
180098E	Escasamente	SI	SI	SI	SI	SI	NO	SI	NO	NO	NO	SI	NO	SI	SI	SI	NO	NO	NO	NO	SI	NO	SI	NO
170721A	Nunca	SI	SI	SI	SI	SI	SI	NO	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	NO	NO	NO
171790G	Siempre	SI	NO	SI	NO	SI	NO	SI	NO	NO	NO	SI	SI	NO	NO	SI	NO	SI	NO	NO	SI	NO	NO	SI
171791C	Ocasionalmente	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	NO	SI	NO
162397D	Frecuentemente	SI	SI	NO	SI	NO	NO	SI	SI	SI	NO	SI	NO	SI	NO	SI	SI	SI	NO	NO	SI	NO	SI	NO
179031H	Escasamente	SI	SI	SI	SI	SI	SI	SI	SI	NO	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	NO	NO	NO
171792J	Frecuentemente	SI	SI	SI	SI	NO	NO	SI	SI	SI	NO	SI	SI	SI	SI	SI	SI	SI	NO	NO	SI	SI	SI	NO
175578B	Frecuentemente	SI	SI	SI	SI	SI	SI	NO	SI	NO	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	NO	NO	NO
162392B	Escasamente	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	NO	NO	SI	SI	SI	SI	NO	NO	NO	SI	NO	NO	NO
162391F	Frecuentemente	SI	SI	NO	SI	SI	SI	SI	SI	SI	NO	SI	SI	SI	NO	SI	SI	NO	NO	NO	SI	NO	NO	SI
170156B	Escasamente	SI	NO	SI	SI	SI	SI	SI	SI	NO	NO	SI	SI	SI	SI	SI	SI	SI	NO	NO	SI	NO	NO	NO
171794B	Frecuentemente	SI	SI	NO	SI	NO	SI	SI	SI	SI	SI	SI	SI	NO	SI	SI	SI	NO	NO	NO	SI	SI	NO	NO
170165 ^a	Frecuentemente	SI	SI	SI	SI	SI	NO	NO	SI	SI	NO	SI	NO	SI	SI	NO	SI	NO	NO	NO	SI	NO	NO	NO
175029I	Ocasionalmente	SI	SI	NO	SI	SI	SI	SI	SI	NO	SI	SI	SI	SI	SI	SI	SI	NO	SI	NO	SI	NO	SI	NO
162395A	Frecuentemente	SI	SI	SI	SI	SI	SI	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	NO	SI	NO	NO	NO

Fuente: Elaboración propia

Continuación de la Tabla 27, Respuestas de las personas encuestadas

171796E	Frecuentemente	SI	SI	NO	SI	SI	SI	SI	SI	NO	NO	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	NO	SI	NO
162420F	Escasamente	SI	NO	SI	SI	SI	SI	NO	SI	NO	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	NO	NO	SI
175024G	Frecuentemente	SI	SI	SI	SI	SI	SI	SI	SI	NO	NO	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	NO	NO	NO
170207F	Ocasionalmente	SI	SI	SI	SI	NO	NO	SI	NO	NO	SI	SI	SI	NO	SI	SI	SI	SI	NO	NO	SI	NO	NO	NO
150312I	Frecuentemente	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO	SI	NO	SI	SI	NO	SI	NO	NO	NO	SI	NO	NO	NO
160265C	Ocasionalmente	SI	SI	SI	SI	SI	SI	SI	SI	NO	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	SI	NO	NO
162480I	Frecuentemente	SI	SI	SI	SI	NO	SI	SI	SI	SI	NO	SI	SI	SI	SI	NO	SI	NO	NO	NO	SI	NO	SI	NO
141867A	Nunca	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI
171797A	Frecuentemente	SI	NO	SI	SI	SI	SI	SI	SI	NO	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	NO	NO	NO
171798H	Ocasionalmente	SI	SI	NO	SI	SI	SI	SI	NO	NO	SI	SI	SI	SI	NO	SI	SI	NO	NO	NO	SI	NO	SI	NO
162473B	Frecuentemente	SI	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	SI	SI	NO	SI	NO	NO	NO	SI	NO	NO	NO
171799D	Frecuentemente	SI	SI	NO	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO	NO	SI	NO	SI	NO
170211C	Escasamente	SI	SI	SI	SI	SI	SI	NO	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO	NO	SI	SI	NO	NO
170212J	Frecuentemente	SI	SI	NO	SI	SI	SI	SI	SI	NO	NO	SI	SI	NO	SI	NO	NO	NO	NO	NO	SI	NO	SI	SI
175581C	Ocasionalmente	SI	SI	SI	SI	SI	SI	SI	NO	SI	NO	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	NO	NO	NO
160274B	Frecuentemente	SI	NO	SI	SI	NO	NO	SI	SI	NO	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	SI	NO	NO	NO
165076D	Nunca	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI
170303E	Frecuentemente	SI	SI	NO	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO	NO	NO	NO	SI	NO	SI	NO
171800B	Frecuentemente	SI	SI	SI	SI	NO	NO	SI	SI	NO	NO	SI	SI	SI	SI	SI	SI	SI	NO	NO	SI	SI	SI	SI
160277A	Ocasionalmente	SI	SI	NO	SI	SI	SI	SI	NO	SI	SI	NO	SI	SI	SI	NO	NO	NO	NO	NO	SI	NO	SI	NO

Fuente: Elaboración propia

Anexo 2: Experimento para medir el tiempo en la búsqueda de las oficinas de la Universidad Nacional Pedro Ruiz Gallo

Para determinar el tiempo perdido por los estudiantes al buscar una oficina de la Universidad Nacional Pedro Ruiz Gallo se ha realizado el siguiente experimento:

El cual consiste en medir el tiempo de 10 participantes que desconocen el campus universitario. De manera individual se calcula el tiempo que tardan en ubicar y recorrer hasta la Facultad de Ciencias Físicas y Matemáticas desde la puerta principal de la universidad. Hemos obtenido los siguientes resultados:

Tiempo Experimental:

Tabla 28.

Tiempo Experimental

Participantes	Tiempo Experimental
Persona 1	4 min 16 seg
Persona 2	5 min 45 seg
Persona 3	6 min 08 seg
Persona 4	5 min 00 seg
Persona 5	5 min 20 seg
Persona 6	7 min 30 seg
Persona 7	5 min 21 seg
Persona 8	5 min 30 seg
Persona 9	5 min 50 seg
Persona 10	5 min 40 seg
Tiempo Promedio	5 min 29 seg

Fuente: Elaboración propia.

Tiempo Control:

Tabla 29.

Tiempo Control

Participante	Tiempo Control
Investigador	3 min 55 seg

Fuente: Elaboración propia.

Los participantes demostraron una tardanza, sin excepción. Comparado con el desplazamiento del investigador, que sí conocen la ubicación de la facultad, se estableció un promedio de tardanza que tienen los participantes, el cual es el 40% de tiempo excedido.



Figura 58: Plano de La Universidad Nacional Pedro Ruíz Gallo
Fuente: (Ugaz Medina, 2009, Levantamiento Topográfico Ciudad Universitaria)

Anexo 4: Implementacion del Grafo que representa las vias Transitables de la Universidad Nacional Pedro Ruíz Gallo



Figura 59: Grafo que representa la Universidad Nacional Pedro Ruíz Gallo
Fuente: Elaboración Propia

Descripción Figura 59:
“El grafo está conformado por 435 nodos y 545 aristas”

Anexo 5: Pruebas realizadas al software

Tabla 30.

Plantilla de prueba de caja negra

Responsable:					
Código prueba:		Fecha:		Modulo:	
Datos de prueba:					
Variables	Clase de equivalencia	Entradas controladas	Salidas esperadas	Salidas obtenida	¿Pruebas superada?
Observación:					

Fuente: Elaboración propia

Tabla 31.
Resultados de las pruebas de caja negra

N°	Código	Fecha	Modulo	Variables	Clase de equivalencia	Entradas controladas	Salidas esperadas	Salidas obtenida	¿Prueba superada?	Observación
1	PU-001	25/03/2019	Usuario	Script Json	valor no numérico	nodo	“Posicionamiento de los nodos”	“Posicionamiento de los nodos”	SI	Prueba realizada con nodos registrados y listados en la API
2	PU-001	25/03/2019	Usuario	Script Json	valor no numérico	arista	“Conexión de los nodos”	“Conexión de los nodos”	SI	Prueba realizada con aristas registradas y listadas en la API
3	PU-001	25/03/2019	Usuario	Script Json	valor no numérico	destino	“listado de destino”	“listado de destino”	SI	Prueba realizada con destinos registrados y listados en la API
4	PU-002	25/03/2019	Usuario	Origen	valor no numérico	Facultad de Ciencias Económicas	“ Origen Seleccionado”	“ Origen Seleccionado”	SI	Prueba realizada con destinos registrados
5	PU-002	25/03/2019	Usuario	Origen	valor no numérico	Facultad de Ciencias Físicas y Matemáticas	“ Origen Seleccionado”	“ Origen Seleccionado”	SI	Prueba realizada con destinos registrados
6	PU-002	25/03/2019	Usuario	Origen	valor no numérico	Facultad de Agronomía	“ Origen Seleccionado”	“ Origen Seleccionado”	SI	Prueba realizada con destinos registrados
7	PU-002	25/03/2019	Usuario	Origen	valor no numérico	Facultad de Enfermería	“ Origen Seleccionado”	“ Origen Seleccionado”	SI	Prueba realizada con destinos registrados
8	PU-003	25/03/2019	Usuario	Destino	valor no numérico	Facultad de Zootecnia	“ Destino Seleccionado”	“ Destino Seleccionado”	SI	Prueba realizada con destinos registrados
9	PU-003	25/03/2019	Usuario	Destino	valor no numérico	Facultad de Medicina Veterinaria	“ Destino Seleccionado”	“ Destino Seleccionado”	SI	Prueba realizada con destinos registrados
10	PU-003	25/03/2019	Usuario	Destino	valor no numérico	Facultad de Medicina Humana	“ Destino Seleccionado”	“ Destino Seleccionado”	SI	Prueba realizada con destinos registrados
11	PU-003	25/03/2019	Usuario	Destino	valor no numérico	Facultad de Ciencias Económicas	“ Destino Seleccionado”	“ Destino Seleccionado”	SI	Prueba realizada con destinos registrados

Fuente: Elaboración propia

Continuación de la Tabla 31, Resultados de las pruebas de caja negra

12	PU-004	25/03/2019	Usuario	Longitud origen	-6.709000 <= n <= -6.705400	-6.706870	Distancia de la ruta Fac. Ciencias Económicas – Fac. Zootecnia : 173.41 m Nodos:15	Distancia de la ruta Fac. Ciencias Económicas – Fac. Zootecnia : 173.41 m Nodos:15	SI	Prueba realizada con nodos, arietas y destinos registrados
				Latitud origen	-79.903520 <= n <= -79.910160	79.908816				
				Longitud destino	-6.709000 <= n <= -6.705400	-6.707873				
				Latitud destino	-79.903520 <= n <= -79.910160	79.909245				
13	PU-004	25/03/2019	Usuario	Longitud origen	-6.709000 <= n <= -6.705400	-6.708091	Distancia de la ruta Fac. Ciencias Físicas y Matemáticas – Fac. Medicina Veterinaria : 291.80 m Nodos: 20	Distancia de la ruta Fac. Ciencias Físicas y Matemáticas – Fac. Medicina Veterinaria : 291.80 m Nodos: 20	SI	Prueba realizada con nodos, arietas y destinos registrados
				Latitud origen	-79.903520 <= n <= -79.910160	79.908545				
				Longitud destino	-6.709000 <= n <= -6.705400	-6.706813				
				Latitud destino	-79.903520 <= n <= -79.910160	79.909723				
14	PU-004	25/03/2019	Usuario	Longitud origen	-6.709000 <= n <= -6.705400	-6.707532	Distancia de la ruta Fac. Agronomía – Fac. Medicina Humana: 305.35 m Nodos: 24	Distancia de la ruta Fac. Agronomía – Fac. Medicina Humana: 305.35 m Nodos: 24	SI	Prueba realizada con nodos, arietas y destinos registrados
				Latitud origen	-79.903520 <= n <= -79.910160	79.908929				
				Longitud destino	-6.709000 <= n <= -6.705400	-6.705691				
				Latitud destino	-79.903520 <= n <= -79.910160	79.907678				
15	PU-004	25/03/2019	Usuario	Longitud origen	-6.709000 <= n <= -6.705400	-6.706691	Distancia de la ruta Fac. Enfermería – Fac. Ciencias Económicas: 86.107 m Nodos: 10	Distancia de la ruta Fac. Enfermería – Fac. Ciencias Económicas: 86.107 m Nodos: 10	SI	Prueba realizada con nodos, arietas y destinos registrados
				Latitud origen	-79.903520 <= n <= -79.910160	79.908472				
				Longitud destino	-6.709000 <= n <= -6.705400	-6.706870				
				Latitud destino	-79.903520 <= n <= -79.910160	79.908816				

Fuente: Elaboración propia

Continuación de la Tabla 31, Resultados de las pruebas de caja negra

16	PU-005	17/03/2019	Administración	Usuario	valor no numérico	Admin	Mensaje "usuario o contraseña incorrecta"	Mensaje "usuario o contraseña incorrecta"	SI	Prueba con una contraseña vacía de un usuario registrado.
				Contraseña	valor no numérico	N/A				
17	PU-005	17/03/2019	Administración	Usuario	valor no numérico	Admin	Mensaje "usuario o contraseña incorrecta"	Mensaje "usuario o contraseña incorrecta"	SI	Prueba con una contraseña Incorrecta de un usuario registrado.
				Contraseña	valor no numérico	123548				
18	PU-006	17/03/2019	Administración	Usuario	valor no numérico	N/A	Mensaje "usuario o contraseña incorrecta"	Mensaje "usuario o contraseña incorrecta"	SI	Prueba con un Id vacía de un usuario registrado.
				Contraseña	valor no numérico	Admin				
19	PU-006	17/03/2019	Administración	Usuario	valor no numérico	Admin1254	Mensaje "usuario o contraseña incorrecta"	Mensaje "usuario o contraseña incorrecta"	SI	Prueba con un Id incorrecto a de un usuario registrado.
				Contraseña	valor no numérico	Admin				
20	PU-007	17/03/2019	Administración	Usuario	valor no numérico	DanteRamLop	Mensaje "Usuario Creado"	Mensaje "Usuario Creado"	SI	Prueba con una contraseña vacía de un usuario registrado.
				Contraseña	valor no numérico	5abril1993				
				Nombre	valor no numérico	Dante				
				Apellido Paterno	valor no numérico	Ramírez				
				Apellido Materno	valor no numérico	López				
				Correo	valor no numérico	danterl@gmail.com				
				Tipo	[0,1]	0				
21	PU-007	17/03/2019	Administración	Usuario	valor no numérico	JeffersonQuiDav	Mensaje "Usuario Creado"	Mensaje "Usuario Creado"	SI	Prueba con un usuario no registrado
				Contraseña	valor no numérico	12marzo1993				
				Nombre	valor no numérico	Jefferson				
				Apellido Paterno	valor no numérico	Quiroz				
				Apellido Materno	valor no numérico	Davila				
				Correo	valor no numérico	Jeff_17@gmail.com				
				Tipo	[0,1]	0				
22	PU-007	17/03/2019	Administración	Usuario	valor no numérico	MariaBelenRod	Mensaje "Usuario Creado"	Mensaje "Usuario Creado"	SI	Prueba con un usuario no registrado
				Contraseña	valor no numérico	2451 mayo				
				Nombre	valor no numérico	María				
				Apellido Paterno	valor no numérico	Rodríguez				
				Apellido Materno	valor no numérico	Chávez				
				Correo	valor no numérico	Belen_1995@gmail.com				
				Tipo	[0,1]	0				

Fuente: Elaboración propia

Continuación de la Tabla 31, Resultados de las pruebas de caja negra

23	PU-007	17/03/2019	Administración	Usuario Contraseña Nombre Apellido Paterno Apellido Materno Correo	valor no numérico valor no numérico valor no numérico valor no numérico valor no numérico	JuanaPeñaLo Mirlo1542 Juana Peña Villalobos VillalobosJuana@gmail.com	Mensaje “Usuario Creado”	Mensaje “Usuario Creado”	SI	Prueba con un usuario no registrado
24	PU-008	17/03/2019	Administración	Tipo Usuario Contraseña Nombre Apellido Paterno Apellido Materno Correo	[0,1] valor no numérico valor no numérico valor no numérico valor no numérico valor no numérico	0 DanteRamLop Juan123cl5 Juan Paz Pérez Juanpaz_5@gmail.com	Mensaje “Usuario Existente”	Mensaje “Usuario Existente”	SI	Prueba con un usuario registrado
25	PU-008	17/03/2019	Administración	Tipo Usuario Contraseña Nombre Apellido Paterno Apellido Materno Correo	[0,1] valor no numérico valor no numérico valor no numérico valor no numérico valor no numérico	0 JeffersonQuiDav TG231244 Jefferson Quiroz Davila Jeff_17@gmail.com	Mensaje “Usuario Existente”	Mensaje “Usuario Existente”	SI	Prueba con un usuario registrado
26	PU-009	17/03/2019	Administración	Tipo Usuario Contraseña Nombre Apellido Paterno Apellido Materno Correo	[0,1] valor no numérico valor no numérico valor no numérico valor no numérico valor no numérico	1 DanteRamLop 1993Ramirez5 Dante Ramírez López danterl@gmail.com	Mensaje “Usuario Modificado”	Mensaje “Usuario Modificado”	SI	Prueba con un usuario registrado
27	PU-009	17/03/2019	Administración	Tipo Usuario Contraseña Nombre Apellido Paterno Apellido Materno Correo	[0,1] valor no numérico valor no numérico valor no numérico valor no numérico valor no numérico	1 JeffersonQuiDav 12marzo1993 Jefferson Quiroz Davila Jeff_17_5@gmail.com	Mensaje “Usuario Modificado”	Mensaje “Usuario Modificado”	SI	Prueba con un usuario registrado
28	PU-010	17/03/2019	Administración	Tipo Usuario Contraseña Nombre Apellido Paterno Apellido Materno Correo	[0,1] valor no numérico valor no numérico valor no numérico valor no numérico valor no numérico	1 N/A 12juan1923 Juan Paz Pérez Juanpaz_5@gmail.com	Mensaje “se deben completar Todos los campos”	Mensaje “se deben completar Todos los campos”	SI	Prueba con un usuario no registrado

Fuente: Elaboración propia

Continuación de la Tabla 31, Resultados de las pruebas de caja negra

29	PU-010	17/03/2019	Administración	Usuario Contraseña Nombre Apellido Paterno Apellido Materno Correo	valor no numérico valor no numérico valor no numérico valor no numérico valor no numérico	JuanPazPer N/A Juan Paz Pérez Juanpaz_5@gmail.com	Mensaje “se deben completar Todos los campos”	Mensaje “se deben completar Todos los campos”	SI	Prueba con un usuario no registrado
30	PU-010	17/03/2019	Administración	Tipo Usuario Contraseña Nombre Apellido Paterno Apellido Materno Correo	[0,1] valor no numérico valor no numérico valor no numérico valor no numérico valor no numérico	JuanPazPer 12juan1923 N/A Paz Pérez Juanpaz_5@gmail.com	Mensaje “se deben completar Todos los campos”	Mensaje “se deben completar Todos los campos”	SI	Prueba con un usuario no registrado
31	PU-010	17/03/2019	Administración	Tipo Usuario Contraseña Nombre Apellido Paterno Apellido Materno Correo	[0,1] valor no numérico valor no numérico valor no numérico valor no numérico valor no numérico	JuanPazPer 12juan1923 Juan N/A Pérez Juanpaz_5@gmail.com	Mensaje “se deben completar Todos los campos”	Mensaje “se deben completar Todos los campos”	SI	Prueba con un usuario no registrado
32	PU-010	17/03/2019	Administración	Tipo Usuario Contraseña Nombre Apellido Paterno Apellido Materno Correo	[0,1] valor no numérico valor no numérico valor no numérico valor no numérico valor no numérico	JuanPazPer 12juan1923 Juan Paz N/A Juanpaz_5@gmail.com	Mensaje “se deben completar Todos los campos”	Mensaje “se deben completar Todos los campos”	SI	Prueba con un usuario no registrado
33	PU-010	17/03/2019	Administración	Tipo Usuario Contraseña Nombre Apellido Paterno Apellido Materno Correo	[0,1] valor no numérico valor no numérico valor no numérico valor no numérico valor no numérico	JuanPazPer 12juan1923 N/A Paz Pérez N/A	Mensaje “se deben completar Todos los campos”	Mensaje “se deben completar Todos los campos”	SI	Prueba con un usuario no registrado
34	PU-011	17/03/2019	Administración	Tipo Usuario	[0,1] valor no numérico	JuanPazPer	Mensaje “Usuario Eliminado”	Mensaje “Usuario Eliminado”	SI	Prueba con un usuario registrado
35	PU-011	17/03/2019	Administración	Usuario	valor no numérico	JuanaPeñaLo	Mensaje “Usuario Eliminado”	Mensaje “Usuario Eliminado”	SI	Prueba con un usuario registrado

Fuente: Elaboración propia

Continuación de la Tabla 31, Resultados de las pruebas de caja negra

36	PU-012	25/03/2019	Administración	Nombre Longitud	valor no numérico -6.709000 <= n <= -6.705400	Nodo 1 -6.706365	Mensaje “Nodo Registrado”	Mensaje “Nodo Registrado”	SI	Prueba con un nodo no registrado
				Latitud	-79.903520 <= n <= -79.910160	-79.906626				
37	PU-012	25/03/2019	Administración	Vinculo Nombre Longitud	0 < n < 7 valor no numérico -6.709000 <= n <= -6.705400	1 Nodo 2 -6.706455	Mensaje “Nodo Registrado”	Mensaje “Nodo Registrado”	SI	Prueba con un nodo no registrado
				Latitud	-79.903520 <= n <= -79.910160	-79.90682				
38	PU-012	25/03/2019	Administración	Vinculo Nombre Longitud	0 < n < 7 valor no numérico -6.709000 <= n <= -6.705400	2 Nodo 3 -6.706127	Mensaje “Nodo Registrado”	Mensaje “Nodo Registrado”	SI	Prueba con un nodo no registrado
				Latitud	-79.903520 <= n <= -79.910160	-79.906626				
39	PU-012	25/03/2019	Administración	Vinculo Nombre Longitud	0 < n < 7 valor no numérico -6.709000 <= n <= -6.705400	3 Nodo 4 -6.706726	Mensaje “Nodo Registrado”	Mensaje “Nodo Registrado”	SI	Prueba con un nodo no registrado
				Latitud	-79.903520 <= n <= -79.910160	-79.906454				
40	PU-013	25/03/2019	Administración	Vinculo Nombre Longitud	0 < n < 7 valor no numérico -6.709000 <= n <= -6.705400	4 Nodo 3 -6.706110	Mensaje “ longitud modificada”	Mensaje “ longitud modificada”	SI	Prueba con un nodo registrado
				Latitud	-79.903520 <= n <= -79.910160	-79.906626				
41	PU-013	25/03/2019	Administración	Vinculo Nombre Longitud	0 < n < 7 valor no numérico -6.709000 <= n <= -6.705400	3 Nodo 4 -6.706700	Mensaje “ Latitud modificada”	Mensaje “ Latitud modificada”	SI	Prueba con un nodo registrado
				Latitud	-79.903520 <= n <= -79.910160	-79.906454				
42	PU-014	25/03/2019	Administración	Vinculo Nombre Longitud	0 < n < 7 valor no numérico -6.709000 <= n <= -6.705400	4 Nodo 1 -6.706365	Mensaje “Nodo existente”	Mensaje “Nodo existente”	SI	Prueba con un nodo registrado
				Latitud	-79.903520 <= n <= -79.910160	-79.906626				
				Vinculo	0 < n < 7	1				

Fuente: Elaboración propia

Continuación de la Tabla 31, Resultados de las pruebas de caja negra

43	PU-014	25/03/2019	Administración	Nombre Longitud	valor no numérico -6.709000 <= n <= -6.705400	Nodo 2 -6.706455	Mensaje “Nodo existente”	Mensaje “Nodo existente”	SI	Prueba con un nodo registrado
				Latitud	-79.903520 <= n <= -79.910160	-79.90682				
				Vinculo	0 < n < 7	2				
44	PU-015	25/03/2019	Administración	Nombre Longitud	valor no numérico -6.709000 <= n <= -6.705400	N/A -6.706945	Mensaje “se deben completar Todos los campos”	Mensaje “se deben completar Todos los campos”	SI	Prueba con un nodo no registrado
				Latitud	-79.903520 <= n <= -79.910160	-79.90064				
				Vinculo	0 < n < 7	5				
45	PU-015	25/03/2019	Administración	Nombre Longitud	valor no numérico -6.709000 <= n <= -6.705400	Nodo 5 N/A	Mensaje “se deben completar Todos los campos”	Mensaje “se deben completar Todos los campos”	SI	Prueba con un nodo no registrado
				Latitud	-79.903520 <= n <= -79.910160	-79.90064				
				Vinculo	0 < n < 7	5				
46	PU-015	25/03/2019	Administración	Nombre Longitud	valor no numérico -6.709000 <= n <= -6.705400	Nodo 5 -6.706945	Mensaje “se deben completar Todos los campos”	Mensaje “se deben completar Todos los campos”	SI	Prueba con un nodo no registrado
				Latitud	-79.903520 <= n <= -79.910160	N/A				
				Vinculo	0 < n < 7	5				
47	PU-016	25/03/2019	Administración	Nombre	valor no numérico	Nodo 3	Mensaje “Nodo Eliminado”	Mensaje “Nodo Eliminado”	SI	Prueba con un nodo registrado Sin vinculo
48	PU-016	25/03/2019	Administración	Nombre	valor no numérico	Nodo 4	Mensaje “Nodo Eliminado”	Mensaje “Nodo Eliminado”	SI	Prueba con un nodo registrado Sin vinculo
49	PU-017	25/03/2019	Administración	Nombre	valor no numérico	Facultad de Agronomía	Mensaje “Destino creado”	Mensaje “Destino creado”	SI	Prueba con un destino no registrado
				Descripción	valor no numérico	Detalles de la oficina:.. Imagen1.jpg				
				Imagen Nodo	Archivo.jpg valor no numérico	Nodo 1				
50	PU-017	25/03/2019	Administración	Nombre	valor no numérico	Facultad de Ciencias Biológicas	Mensaje “Destino creado”	Mensaje “Destino creado”	SI	Prueba con un destino no registrado
				Descripción	valor no numérico	Detalles de la oficina:.. Imagen2.jpg				
				Imagen Nodo	Archivo.jpg valor no numérico	Nodo 2				

Fuente: Elaboración propia

Continuación de la Tabla 31, Resultados de las pruebas de caja negra

51	PU-017	25/03/2019	Administración	Nombre	valor no numérico	Facultad de Ciencias Económicas, Administrativas y Contables	Mensaje “Destino creado”	Mensaje “Destino creado”	SI	Prueba con un destino no registrado
				Descripción	valor no numérico	Detalles de la oficina:..				
				Imagen	Archivo jpg	Imagen3.jpg				
52	PU-017	25/03/2019	Administración	Nombre	valor no numérico	Nodo 3	Mensaje “Destino creado”	Mensaje “Destino creado”	SI	Prueba con un destino no registrado
				Descripción	valor no numérico	Facultad de Ciencias Físicas y Matemáticas				
				Imagen	Archivo jpg	Detalles de la oficina:..				
				Nodo	Imagen4.jpg					
53	PU-018	25/03/2019	Administración	Nombre	valor no numérico	Nodo 4	Mensaje “Destino Modificado”	Mensaje “Destino Modificado”	SI	Prueba con un destino registrado
				Descripción	valor no numérico	Facultad de Ciencias Económicas, Administrativas y Contables				
				Imagen	Archivo jpg	Detalles de la oficina:..				
				Nodo	Imagen5.jpg					
54	PU-018	25/03/2019	Administración	Nombre	valor no numérico	Nodo 5	Mensaje “Destino Modificado”	Mensaje “Destino Modificado”	SI	Prueba con un destino registrado
				Descripción	valor no numérico	Facultad de Ciencias Físicas y Matemáticas				
				Imagen	Archivo jpg	Detalles de la oficina:..				
				Nodo	Imagen6.jpg					
55	PU-019	25/03/2019	Administración	Nombre	valor no numérico	Nodo 6	Mensaje “Destino Existente”	Mensaje “Destino Existente”	SI	Prueba con un destino registrado
				Descripción	valor no numérico	Facultad de Ciencias Económicas, Administrativas y Contables				
				Imagen	Archivo jpg	Detalles de la oficina:..				
				Nodo	Imagen3.jpg					
56	PU-019	25/03/2019	Administración	Nombre	valor no numérico	Nodo 3	Mensaje “Destino Existente”	Mensaje “Destino Existente”	SI	Prueba con un destino registrado
				Descripción	valor no numérico	Facultad de Ciencias Físicas y Matemáticas				
				Imagen	Archivo jpg	Detalles de la oficina:..				
				Nodo	Imagen4.jpg					
					valor no numérico	Nodo 4				

Fuente: Elaboración propia

Continuación de la Tabla 31, Resultados de las pruebas de caja negra

57	PU-020	25/03/2019	Administración	Nombre Descripción	valor no numérico valor no numérico	N/A Detalles de la oficina:..	Mensaje “se deben completar Todos los campos”	Mensaje “se deben completar Todos los campos”	SI	Prueba con un destino no registrado
58	PU-020	25/03/2019	Administración	Imagen Nodo Nombre	Archivo jpg valor no numérico valor no numérico	Imagen5.jpg Nodo 5 Facultad de Ciencias Histórico Sociales y Educación	Mensaje “se deben completar Todos los campos”	Mensaje “se deben completar Todos los campos”	SI	Prueba con un destino no registrado
59	PU-020	25/03/2019	Administración	Descripción Imagen Nodo Nombre	valor no numérico Archivo jpg valor no numérico valor no numérico	Detalles de la oficina:.. N/A Nodo 5 Facultad de Ciencias Histórico Sociales y Educación	Mensaje “se deben completar Todos los campos”	Mensaje “se deben completar Todos los campos”	SI	Prueba con un destino no registrado
60	PU-021	25/03/2019	Administración	Descripción Imagen Nodo Nombre	valor no numérico Archivo jpg valor no numérico valor no numérico	Detalles de la oficina:.. Imagen5.jpg N/A Facultad de Agronomía	Mensaje “se deben completar Todos los campos”	Mensaje “se deben completar Todos los campos”	SI	Prueba con un destino registrado
61	PU-021	25/03/2019	Administración	Nombre	valor no numérico	Facultad de Ciencias Biológicas	Mensaje “Destino Eliminado” Mensaje “Destino Eliminado”	Mensaje “Destino Eliminado” Mensaje “Destino Eliminado”	SI	Prueba con un destino registrado
62	PU-022	25/03/2019	Administración	Nodo 1 Nodo 2	valor no numérico valor no numérico	Nodo 24 Nodo 25	Mensaje “Arista creada”	Mensaje “Arista creada”	SI	Prueba con un arista no registrada
63	PU-022	25/03/2019	Administración	Nodo 1 Nodo 2	valor no numérico valor no numérico	Nodo 86 Nodo 84	Mensaje “Arista creada”	Mensaje “Arista creada”	SI	Prueba con un arista no registrada
64	PU-022	25/03/2019	Administración	Nodo 1 Nodo 2	valor no numérico valor no numérico	Nodo 128 Nodo 215	Mensaje “Arista creada”	Mensaje “Arista creada”	SI	Prueba con un arista no registrada
65	PU-023	25/03/2019	Administración	Nodo 1 Nodo 2	valor no numérico valor no numérico	Nodo 53 Nodo 21	Mensaje “Arista modificada”	Mensaje “Arista modificada”	SI	Prueba con una arista registrada

Fuente: Elaboración propia

Continuación de la Tabla 31, Resultados de las pruebas de caja negra

66	PU-023	25/03/2019	Administración	Nodo 1	valor no numérico	Nodo 55	Mensaje “Arista modificada”	Mensaje “Arista modificada”	SI	Prueba con una arista registrada
				Nodo 2	valor no numérico	Nodo 45				
67	PU-023	25/03/2019	Administración	Nodo 1	valor no numérico	Nodo 78	Mensaje “Arista modificada”	Mensaje “Arista modificada”	SI	Prueba con una arista registrada
				Nodo 2	valor no numérico	Nodo 79				
68	PU-024	25/03/2019	Administración	Nodo 1	valor no numérico	Nodo 16	Mensaje “Arista modificada”	Mensaje “Arista modificada”	SI	Prueba con un arista no registrada
				Nodo 2	valor no numérico	Nodo 17				
69	PU-024	25/03/2019	Administración	Nodo 1	valor no numérico	Nodo 45	Mensaje “Arista modificada”	Mensaje “Arista modificada”	SI	Prueba con un arista no registrada
				Nodo 2	valor no numérico	Nodo 155				
70	PU-024	25/03/2019	Administración	Nodo 1	valor no numérico	Nodo 156	Mensaje “Arista modificada”	Mensaje “Arista modificada”	SI	Prueba con un arista no registrada
				Nodo 2	valor no numérico	Nodo 311				
71	PU-025	25/03/2019	Administración	Nodo 1	valor no numérico	Nodo 312	Mensaje “Arista eliminada”	Mensaje “Arista eliminada”	SI	Prueba con una arista registrada
				Nodo 2	valor no numérico	Nodo 215				
72	PU-025	25/03/2019	Administración	Nodo 1	valor no numérico	Nodo 285	Mensaje “Arista eliminada”	Mensaje “Arista eliminada”	SI	Prueba con una arista registrada
				Nodo 2	valor no numérico	Nodo 284				
73	PU-025	25/03/2019	Administración	Nodo 1	valor no numérico	Nodo 256	Mensaje “Arista eliminada”	Mensaje “Arista eliminada”	SI	Prueba con una arista registrada
				Nodo 2	valor no numérico	Nodo 254				
74	PU-025	25/03/2019	Administración	Nodo 1	valor no numérico	Nodo 165	Mensaje “Arista eliminada”	Mensaje “Arista eliminada”	SI	Prueba con una arista registrada
				Nodo 2	valor no numérico	Nodo 175				
75	PU-026	25/03/2019	Administración	Grafo	valor no numérico	Nodos	“Posicionamient o de los nodos”	“Posicionamient o de los nodos”	SI	Prueba con arista,nodos y destinos registrados
76	PU-026	25/03/2019	Administración	Grafo	valor no numérico	Aristas	“Conexión de los nodos”	“Conexión de los nodos”	SI	Prueba con arista,nodos y destinos registrados
77	PU-026	25/03/2019	Administración	Grafo	valor no numérico	Destinos	“Posicionamient o de los destinos”	“Posicionamient o de los destinos”	SI	Prueba con arista,nodos y destinos registrados
78	PU-027	02/11/2019	Administración	Fecha	dd/mm/aa	01/11/2019	“Número de visitas”	“Número de visitas”	SI	Prueba con una fecha específica
79	PU-027	02/11/2019	Administración	Fecha	dd/mm/aa	02/11/2019	“Número de visitas”	“Número de visitas”	SI	Prueba con una fecha específica

Fuente: Elaboración propia

Anexo 6: Implementación

1) Paquete Vista

a) Script “index.html”

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>UNIMAPS</title>
<link rel="stylesheet" type="text/css" href="css/bootstrap.css" />
<link rel="stylesheet" type="text/css" href="css/style.css" />
<script src="js/jquery-1.11.1.js"></script>
</head>
<body class="login">
<div class="container" style="width:30%; padding-top: 9%;">
<form action="src/ValidarUsuario.php" method="POST">
<h2 class="form-signin-heading">
Iniciar Sesion
</h2>

<label for="txtUsuario" class="sr-only">Login</label>
<input type="text" name="txtUsuario" id="txtUsuario" class="form-control" placeholder=
"Usuario" required autofocus>
<label for="txtClave" class="sr-only">Clave</label>
<input type="password" name="txtClave" id="txtClave" class="form-control" placeholder=
"Clave" required>

<div class="checkbox">
<label>
<input type="checkbox" value="remember-me"> Recuerdame
</label>
</div>
<button class="btn btn-lg btn-primary btn-block" name="LOGIN" value="LOGIN"
type="submit">Login</button>
</form>
</div>
</body>
</html>
```

b) Script “usuario.html”

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Panel de Control Usuario</title>
<link rel="stylesheet" type="text/css" href="css/bootstrap.css" />
<link rel="stylesheet" type="text/css" href="css/style.css" />
<script src="js/jquery-1.11.1.js"></script>
<script src="js/bootstrap.min.js"></script>
</head>
<body>
<div class="container">
<div class="col-md-12"> <!-- menu-->
<ul class="nav nav-tabs nav-justified">
<li class="active"><a href="inicio.php">Inicio</a></li>
<li class="active"><a href="usuario.php">Usuario</a></li>
<li class="active"><a href="nodo.php">Nodo</a></li>
<li class="active"><a href="destino.php">Destino</a></li>
<li class="active"><a href="arista.php">Arista</a></li>
<li class="active"><a href="grafo.html">Grafo</a></li>
<li><a href="cerrarSesion.php">Cerrar Sesion</a></li>
</ul>
```



```

        <li><a href="cerrarSesion.php">Cerrar Sesion</a></li>
    </ul>
</div>
<!-- <div style="width:50%">-->
<div class="col-md-12">
<div class="row">
</div>
<div class="row">
    <div class="col-md-12 text-center">
        <h3 class="page-header">REGISTRAR PERSONA</h3>
    </div>
</div>
<form name="frmUsuario" action="usuario_frm.php" method="POST" class="col-md-8 col-md-offset-2">
    <input type="hidden" name="id" id="id" class="form-control" value="<?php echo $cod_usuario ?>"><!-- -->
    <div class="row">
        <div class="col-md-12 left"><hr></div>
    </div>
    <div class="row">
        <div class="col-md-6 left">
            <div class="form-group" ng-class="{ 'has-error': frm.txtNombres.$invalid, 'has-success': frm.txtNombres.$valid}">
                <label for="txtNombres">NOMBRES</label>
                <input id="txtNombres" value="<?php echo $nombres ?>" type="text" maxlength="30" class="form-control" name="txtNombres" required="" placeholder="" autofocus="">
            </div>
        </div>
        <div class="col-md-6 left">
            <div class="form-group" ng-class="{ 'has-error': frm.txtAp_pat.$invalid, 'has-success': frm.txtAp_pat.$valid}">
                <label for="txtAp_pat">APELLIDO PATERNO</label>
                <input id="txtAp_pat" value="<?php echo $ap_pat ?>" type="text" maxlength="30" class="form-control" name="txtAp_pat" required="" placeholder="">
            </div>
        </div>
    </div>
    <div class="row">
        <div class="col-md-6 left">
            <div class="form-group" ng-class="{ 'has-error': frm.txtAp_mat.$invalid, 'has-success': frm.txtAp_mat.$valid}">
                <label for="txtAp_mat">APELLIDO MATERNO</label>
                <input id="txtAp_mat" value="<?php echo $ap_mat ?>" type="text" maxlength="8" class="form-control" name="txtAp_mat" required="" placeholder="">
            </div>
        </div>
    </div>
    <div class="row">
        <div class="col-md-12 left">
            <div class="form-group" ng-class="{ 'has-error': frm.txtcorreo.$invalid, 'has-success': frm.txtcorreo.$valid}">
                <label for="txtcorreo">CORREO</label>
                <input id="txtcorreo" value="<?php echo $correo ?>" type="text" maxlength="100" class="form-control" name="txtcorreo" required="" placeholder="">
            </div>
        </div>
    </div>
    <div class="row" id="validaroculto">
        <div class="col-md-6 left">
            <div class="form-group"><!-- ng-class="{ 'has-error': frm.txtUsuario.$invalid, 'has-success': frm.txtUsuario.$valid}" -->
                <label for="txtUsuario">USUARIO</label>
                <input id="txtUsuario" value="<?php echo $usuario ?>" type="text" maxlength="100" class="form-control" name="txtUsuario" required="" placeholder="">
            </div><!-- -->
        </div>
        <div class="col-md-6 left">
            <div class="form-group"><!-- ng-class="{ 'has-error': frm.txtClave.$invalid, 'has-success': frm.txtClave.$valid}" -->
                <label for="txtClave">CLAVE</label>
                <input id="txtClave" value="<?php echo $contrasena ?>" type="password" maxlength="100"

```

```

        class="form-control" name="txtClave" required="" placeholder="">
    </div>
</div>
</div>
<div class="row">
    <div class="col-md-12 center">
        <button class="btn btn-success" name="GRABAR" >Guardar</button><!-- ng-
            disabled="frm.$invalid" -->
        <a href="usuario.php" class="btn btn-primary">Cancelar</a>
        <hr>
    </div>
</div>
</form>
</div>
</div>
</body>
</html>

```

d) Script “nodo.html”

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title>Panel de Control Nodo</title>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css" />
    <link rel="stylesheet" type="text/css" href="css/style.css" />
    <script src="js/jquery-1.11.1.js"></script>
    <script src="js/bootstrap.min.js"></script>
</head>
<body>
<div class="container">
    <div class="col-md-12"> <!-- menu-->
        <ul class="nav nav-tabs nav-justified">
            <li class="active"><a href="inicio.php">Inicio</a></li>
            <li class="active"><a href="usuario.php">Usuario</a></li>
            <li class="active"><a href="nodo.php">Nodo</a></li>
            <li class="active"><a href="destino.php">Destino</a></li>
            <li class="active"><a href="arista.php">Arista</a></li>
            <li class="active"><a href="grafo.html">Grafo</a></li>
            <li><a href="cerrarSesion.php">Cerrar Sesion</a></li>
        </ul>
    </div>
</div>
<div class="row">&nbsp;</div>
<div class="panel panel-default">
    <div class="panel-heading">
        <h3>LISTADO DE NODOS<div style="float:right; padding-right:35px">
            <a href="nodo_frm.php"><span class="glyphicon glyphicon-plus" aria-hidden="true">
        </span></a></h3>
    </div>
</div>
<table class="table table-hover">
<thead>
<tr>
    <th>NOMBRE</th>
    <th>COORDENADA X</th>
    <th>COORDENADA Y</th>
    <th>USUARIO</th>
    <th>ESTADO</th>
    <th>ACCION</th>
</tr>
</thead>
<tbody>
<?php
    $i=1;
    while($row = $result->fetch_assoc()){
        ?>
    <tr>

```

```
 <?php echo $row["n_nombre"] ?></td>  <?php echo $row["coordenada_x"] ?></td>  <?php echo $row["coordenada_y"] ?></td>  <?php echo $row["u_nombre"] ?></td>  <?php echo $row["estado"] ?></td> <th>     <a href="nodo_frm.php?cod_nodo=<?php echo $row["cod_nodo"]; ?>"><span class=         "glyphicon glyphicon-pencil" aria-hidden="true"></span></a>     &nbsp;     <a href="nodo.php?cod_nodo=<?php echo $row["cod_nodo"]; ?>"><span class="glyphicon         glyphicon-remove" aria-hidden="true"></span></a> </th> </tr> <?php } ?> </tbody> </table> </div> </div> </body> </html> | | | | |
```

e) Script “nodo_frm.html”

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<link rel="stylesheet" type="text/css" href="css/bootstrap.css" />
<link rel="stylesheet" type="text/css" href="css/style.css" />
<script src="js/jquery-1.11.1.js"></script>
<script src="js/bootstrap.min.js"></script>
</head>
<body>
<div class="container">
<div class="col-md-12"> <!-- menu-->
<ul class="nav nav-tabs nav-justified">
<li class="active"><a href="inicio.php">Inicio</a></li>
<li class="active"><a href="usuario.php">Usuario</a></li>
<li class="active"><a href="nodo.php">Nodo</a></li>
<li class="active"><a href="destino.php">Destino</a></li>
<li class="active"><a href="arista.php">Arista</a></li>
<li><a href="cerrarSesion.php">Cerrar Sesion</a></li>
</ul>
</div>
<!-- <div style="width:50%">-->
<div class="col-md-12">
<div class="row">
<div class="row">
<div class="col-md-12 text-center ">
<h3 class="page-header">REGISTRAR NODO</h3>
</div>
</div>
<form name="frmNodo" action="nodo_frm.php" method="POST" class="col-md-8 col-md-offset-2 ">
<input type="hidden" name="id" id="id" class="form-control" value="<?php echo $cod_nodo ?>">
<br>
</div>
<br>
<div class="row">
<div class="col-md-6 left">
<div class="form-group" ng-class="{ 'has-error': frm.txtn_nombres.$invalid, 'has-success':
    frm.txtn_nombre.$valid}">
<label for="txtn_nombre">NOMBRE DEL NODO</label>
<input id="txtn_nombre" value="<?php echo $n_nombre ?>" type="text" maxlength="30"
    class="form-control" name="txtn_nombre" required="" placeholder="" autofocus="">
</div>
</div>
<div class="col-md-6 left">

```

```

<div class="form-group" ng-class="{ 'has-error': frm.txtcoordenada_x.$invalid, 'has-success':
    frm.txtcoordenada_x.$valid}">
    <label for="txtcoordenada_x">COORDENADA X</label>
    <input id="txtcoordenada_x" value="<?php echo $coordenada_x ?>" type="text" maxlength="30"
        class="form-control" name="txtcoordenada_x" required="" placeholder="">
</div>
</div>
</div>
<div class="row">
    <div class="col-md-6 left">
        <div class="form-group" ng-class="{ 'has-error': frm.txtcoordenada_y.$invalid,
            'has-success': frm.txtcoordenada_y.$valid}">
            <label for="txtcoordenada_y">COORDENADA Y</label>
            <input id="txtcoordenada_y" value="<?php echo $coordenada_y ?>" type="text" maxlength
                ="30" class="form-control" name="txtcoordenada_y" required="" placeholder=""
                autofocus="">
        </div>
    </div>
</div>
<div class="row">
    <div class="col-md-12 center">
        <button class="btn btn-success" name="GRABAR" >Guardar</button>
        <!-- ng-disabled="frm.$invalid" -->
        <a href="nodo.php" class="btn btn-primary">Cancelar</a>
        <hr>
    </div>
</div>
</form>
</div>
</div>
<?php echo $_SESSION["cod_usuario"] ?>
</body>

```

f) Script “inicio.html”

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title>Panel de Control</title>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css" />
    <link rel="stylesheet" type="text/css" href="css/style.css" />
    <script src="js/jquery-1.11.1.js"></script>
    <script src="js/bootstrap.min.js"></script>
</head>
<body>
<div class="container">
    <div class="col-md-12"> <!-- menu-->
        <ul class="nav nav-tabs nav-justified">
            <li class="active"><a href="inicio.php">Inicio</a></li>
            <li class="active"><a href="usuario.php">Usuario</a></li>
            <li class="active"><a href="nodo.php">Nodo</a></li>
            <li class="active"><a href="destino.php">Destino</a></li>
            <li class="active"><a href="arista.php">Arista</a></li>
            <li class="active"><a href="grafo.html">Grafo</a></li>
            <li><a href="cerrarSesion.php">Cerrar Sesion</a></li>
        </ul>
    </div>
<div class="row">
    <div class="text-center col-md-12" style="color:#827717;">
        <h3>UNIMPAS</h3>
    </div>
    <div class="col-md-12">
        <div class="col-md-6 col-md-offset-3">
            
        </div>
    </div>
<div class="col-md-12 text-center" style="color:#7CB342">

```

```

        <h4>Bienvenido(a) <?php echo $_SESSION["u_nombre"] ?> <?php echo $_SESSION["ap_pat"] ?>
    </h4>
</div>
</div>
</div>
</body>
</html>

```

g) Script “destino.html”

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title>SISTEMA</title>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css" />
    <link rel="stylesheet" type="text/css" href="css/style.css" />
    <script src="js/jquery-1.11.1.js"></script>
    <script src="js/bootstrap.min.js"></script>
</head>
<body>
    <div class="container">
        <div class="col-md-12"> <!-- menu-->
            <ul class="nav nav-tabs nav-justified">
                <li class="active"><a href="inicio.php">Inicio</a></li>
                <li class="active"><a href="usuario.php">Usuario</a></li>
                <li class="active"><a href="nodo.php">Nodo</a></li>
                <li class="active"><a href="destino.php">Destino</a></li>
                <li class="active"><a href="arista.php">Arista</a></li>
                <li class="active"><a href="grafo.html">Grafo</a></li>
                <li><a href="cerrarSesion.php">Cerrar Sesion</a></li>
            </ul>
        </div>
        <div class="row">&nbsp;</div>
        <div class="panel panel-default">
            <div class="panel-heading">
                <h3>LISTADO DE DESTINOS<div style="float:right; padding-right:35px">
                    <a href="destino_frm.php"><span class="glyphicon glyphicon-plus" aria-
                    hidden="true"></span></a></div>
                </h3>
            </div>
            </div>
            <table class="table table-hover">
                <thead>
                    <tr>
                        <th>NOMBRE</th>
                        <th>DESCRIPCION</th>
                        <th>NODO</th>
                        <th>USUARIO</th>
                        <th>IMAGEN</th>
                        <th>ESTADO</th>
                        <th>ACCION</th>
                    </tr>
                </thead>
                <tbody>
                    <?php
                        $i=1;
                        while($row = $result->fetch_assoc()){
                            ?>
                            <tr>
                                <td><?php echo $row["d_nombre"] ?></td>
                                <td><?php echo $row["descripcion"] ?></td>
                                <td><?php echo $row["n_nombre"] ?></td>
                                <td><?php echo $row["u_nombre"] ?></td>
                                <td><?php echo "<img src='".$row["imagen"]." width='100' height='100'" ?></td>
                                <td><?php echo $row["estado"] ?></td>
                                <th>
                                    <a href="destino_frm.php?cod_destino=<?php echo $row["cod_destino"]; ?>"><span
                                    class="glyphicon glyphicon-pencil" aria-hidden="true"></span></a>
                                    &nbsp;  
                                </th>
                            </tr>
                        }
                    </?php>
                </tbody>
            </table>
        </div>
    </div>

```

```

        <a href="destino.php?cod_destino=<?php echo $row["cod_destino"]; ?>"><span
            class="glyphicon glyphicon-remove" aria-hidden="true"></span></a>

        </th>
    </tr>
    <?php } ?>
</tbody>
</table>
</div>
</div>
</body>
</html>

```

h) Script “destino frm.html”

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<link rel="stylesheet" type="text/css" href="css/bootstrap.css" />
<link rel="stylesheet" type="text/css" href="css/style.css" />
<script src="js/jquery-1.11.1.js"></script>
<script src="js/bootstrap.min.js"></script>
</head>
<body>
<div class="container">
<div class="col-md-12"> <!-- menu-->
<ul class="nav nav-tabs nav-justified">
<li class="active"><a href="inicio.php">Inicio</a></li>
<li class="active"><a href="usuario.php">Usuario</a></li>
<li class="active"><a href="nodo.php">Nodo</a></li>
<li class="active"><a href="destino.php">Destino</a></li>
<li class="active"><a href="arista.php">Arista</a></li>
<li><a href="cerrarSesion.php">Cerrar Sesion</a></li>
</ul>
</div>
<div class="col-md-12">
<div class="row">
<div class="col-md-12 text-center">
<h3 class="page-header">REGISTRAR DESTINO</h3>
</div>
</div>
<div class="col-md-8 left">
<div class="form-group"><!-- ng-class="{ 'has-error': frm.txtNombres.$invalid, 'has-success':
frm.txtNombres.$valid}" -->
<label for="txtnomdestino">NOMBRE DEL DESTINO</label>
<input id="txtnomdestino" value="<?php echo $d_nombre ?>" type="text" maxlength="450"
class="form-control" name="txtnomdestino" placeholder="">
</div>
</div>
<div class="col-md-8 left">
<div class="form-group"><!-- ng-class="{ 'has-error': frm.txtNombres.$invalid, 'has-success':
frm.txtNombres.$valid}" -->
<label for="txtdescripcion">DESCRIPCION DEL DESTINO</label>
<input id="txtdescripcion" value="<?php echo $descripcion ?>" type="text" maxlength="450"
class="form-control" name="txtdescripcion" placeholder="">
</div>
<div class="form-group">
<label for="txtimagen">IMAGEN</label><br>
<?php
if ($imagen=='') {
echo "Selecione imagen";?>
<form method="POST" enctype="multipart/form-data"><input type="file" name="foto"
id="foto">
</form><?php
} else {
echo "<img src='".$imagen.'" width='100' height='100'>";?>
<form method="POST" enctype="multipart/form-data"><input type="file" name="foto">

```



```

        id="foto">
    </form><?php
    }
    ?>
</div>
<div class="col-md-12 left">
    <label for="txtnodo">NODO ASIGANDO</label>
    <select id="txtnodo" name="txtnodo" class="form-control"    >
        <!-- onchange="llamar();" -->
        <option value=""></option>
        <?php
            while($nod = $nodo->fetch_assoc()) {
                ?>
                <option value="<?php echo $nod["cod_nodo"];?>" <?php echo ($cod_nodo ==
                    $nod["cod_nodo"]) ? 'selected' : ''; ?>>
                <?php echo $nod["n_nombre"] ?>
            </option>
        <?php }
        ?>
    </select>
</div>
</div>
<form name="frmdestino" action="destino_frm.php" method="POST" class="col-md-8 col-md-
offset-2 ">
    <input type="hidden" name="id" id="id" class="form-control" value="<?php echo
        $cod_nodo ?>">
    <div class="row">
        </div>
        <br>
        <div class="row">
            <div class="col-md-12 center">
                <button class="btn btn-success" name="GRABAR" >Guardar</button><!-- ng-
                    disabled="frm.$invalid" -->
                <a href="servicios.php" class="btn btn-primary">Cancelar</a>
            </div>
        </div>
    </div>
</form>
</div>
</div>
</body>
</html>

```

i) Script “arista.html”

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
        <title>SISTEMA</title>
        <link rel="stylesheet" type="text/css" href="css/bootstrap.css" />
        <link rel="stylesheet" type="text/css" href="css/style.css" />
        <script src="js/jquery-1.11.1.js"></script>
        <script src="js/bootstrap.min.js"></script>
    </head>
    <body>
        <div class="container">
            <div class="col-md-12"> <!-- menu-->
                <ul class="nav nav-tabs nav-justified">
                    <li class="active"><a href="inicio.php">Inicio</a></li>
                    <li class="active"><a href="usuario.php">Usuario</a></li>
                    <li class="active"><a href="nodo.php">Nodo</a></li>
                    <li class="active"><a href="destino.php">Destino</a></li>
                    <li class="active"><a href="arista.php">Arista</a></li>
                    <li class="active"><a href="grafo.html">Grafo</a></li>
                    <li><a href="cerrarSesion.php">Cerrar Sesion</a></li>
                </ul>
            </div>

```

```

<div class="row">&nbsp;</div>
<div class="panel panel-default">
  <div class="panel-heading">
    <h3>LISTADO DE ARISTAS<div style="float:right; padding-right:35px">
      <a href="arista_frm.php"><span class="glyphicon glyphicon-plus" aria-
        hidden="true"></span></a></h3>
    </div>
  </div>
  <table class="table table-hover">
    <thead>
      <tr>
        <th>ARISTA</th>
        <th>NODO 1</th>
        <th>NODO 2</th>
        <th>ACCION</th>
      </tr>
    </thead>
    <tbody>
      <?php
        $i=1;
        while($row = $result->fetch_assoc()){
          ?>
          <tr>
            <td><?php echo $row["cod_arista"] ?></td>
            <td><?php echo $row["nombre1"] ?></td>
            <td><?php echo $row["nombre2"] ?></td>
            <th>
              <a href="arista_frm.php?cod_arista=<?php echo $row["cod_arista"]; ?>"><span
                class="glyphicon glyphicon-pencil" aria-hidden="true"></span></a>
              &nbsp;<
              <a href="arista.php?cod_arista=<?php echo $row["cod_arista"]; ?>"><span
                class="glyphicon glyphicon-remove" aria-hidden="true"></span></a>
            </th>
          </tr>
          <?php } ?>
        </tbody>
      </table>
    </div>
  </div>
</body>
</html>

```

j) Script “arista_frm.html”

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<link rel="stylesheet" type="text/css" href="css/bootstrap.css" />
<link rel="stylesheet" type="text/css" href="css/style.css" />
<script src="js/jquery-1.11.1.js"></script>
<script src="js/bootstrap.min.js"></script>
</head>
<body>
  <div class="container">
    <div class="col-md-12"> <!-- menu-->
      <ul class="nav nav-tabs nav-justified">
        <li class="active"><a href="inicio.php">Inicio</a></li>
        <li class="active"><a href="usuario.php">Usuario</a></li>
        <li class="active"><a href="nodo.php">Nodo</a></li>
        <li class="active"><a href="destino.php">Destino</a></li>
        <li class="active"><a href="arista.php">Arista</a></li>
        <li><a href="cerrarSesion.php">Cerrar Sesion</a></li>
      </ul>
    </div>
    <div class="col-md-12">
      <div class="row">

```

```

</div>
<div class="row">
  <div class="col-md-12 text-center ">
    <h3 class="page-header">REGISTRAR ARISTA</h3>
  </div>
</div>
<form name="frmarista" action="arista_frm.php" method="POST" class="col-md-8 col-md-
  offset-2 ">
  <input type="hidden" name="id" id="id" class="form-control" value="<?php echo
    $cod_arista ?>">
  <br>
  <div class="col-md-12 left">
    <label for="txtnodoi">NODO DE INICIO</label>
    <select id="txtnodoi" name="txtnodoi" class="form-control" >
      <option value=""></option>
    <?php
      while($nod = $nodo1->fetch_assoc()) {
        ?>
        <option value="<?php echo $nod["cod_nodo"];?>" <?php echo ($cod_nodo_1 ==
          $nod["cod_nodo"]) ? 'selected' : ''; ?>>
          <?php echo $nod["n_nombre"] ?>
        </option>
      <?php
      }
    <?>
    </select>
  </div>
  <br>
  <div class="col-md-12 left">
    <label for="txtnodof">NODO FINAL</label>
    <select id="txtnodof" name="txtnodof" class="form-control" >
      <option value=""></option>
    <?php
      while($nod = $nodo2->fetch_assoc()) {
        ?>
        <option value="<?php echo $nod["cod_nodo"];?>" <?php echo ($cod_nodo_2 ==
          $nod["cod_nodo"]) ? 'selected' : ''; ?>>
          <?php echo $nod["n_nombre"] ?>
        </option>
      <?php
      }
    <?>
    </select>
  </div>
</div>
  <div>
    <br>
    <div class="row">
      <br>
      <div class="col-md-12 center">
        <button class="btn btn-success" name="GRABAR" >Guardar</button><!--ng
          disabled="frm.$invalid" -->
        <a href="arista.php" class="btn btn-primary">Cancelar</a>
      </div>
    </div>
    <?php echo $cod_arista;?>
  </form>
</div>
</div>
</body>
</html>

```

k) Script “grafo.html”

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Mapa UNPRG</title>

```

```

<link rel="stylesheet" type="text/css" href="css/estilo.css">
<link rel="stylesheet" type="text/css" href="css/bootstrap.css" />
<link rel="stylesheet" type="text/css" href="css/style.css" />
<script src="js/jquery-1.11.1.js"></script>
<script src="js/bootstrap.min.js"></script>
</head>
<body>
<div class="col-md-12"> <!-- menu-->
<ul class="nav nav-tabs nav-justified">
<li class="active"><a href="inicio.php">Inicio</a></li>
<li class="active"><a href="usuario.php">Usuario</a></li>
<li class="active"><a href="nodo.php">Nodo</a></li>
<li class="active"><a href="destino.php">Destino</a></li>
<li class="active"><a href="arista.php">Arista</a></li>
<li class="active"><a href="grafo.html">Grafo</a></li>
<li><a href="cerrarSesion.php">Cerrar Sesion</a></li>
</ul>
</div>

<h2>Generar Grafo</h2>
<form action="ruta.php" method="post" enctype="multipart/form-data">

<label>Cargar Lienzo:</label>
<input type="file" name="imagen" required>
<label>Confirmacion:</label>
<input type="submit" value="Enviar">

<div id="contenedor">
<canvas id="lienzo" width="4200px" height="2700px" style="border: 1px solid #000000">
</canvas>
</div>
</form>
<script src="js/canvas.js"></script>
</body>
</html>

```

2) Paquete Modelo

a) Script “arista.php”

```

<?php
include("../system/src/conexion.php");

if (! empty($_GET["cod_arista"])){
    $cod_destino = $_GET["cod_arista"];
    if($cod_destino > 0){
        $sql = "update arista set estado = 0 where cod_arista=".$_cod_arista;
        $res=$conn->query($sql);
        if ($res == 1 ) {
            header("location: arista.php");
        }
    }
}

$sql = " select a.cod_arista, n1.cod_nodo, n1.n_nombre as nombre1, n2.cod_nodo, n2.n_nombre
as nombre2 FROM arista a INNER JOIN nodo n1 ON a.cod_nodo_1= n1.cod_nodo INNER JOIN
nodo n2 ON a.cod_nodo_2 = n2.cod_nodo where a.estado = 1";
$result=$conn->query($sql);
?>

```

b) Script “arista_frm.php”

```

<?php
include("../system/src/conexion.php");

$res="";
$cod_arista = 0;
$cod_nodo_1 = 0;

```

```

$cod_nodo_2 = 0;

if (! empty($_GET["cod_arista"])){
    $cod_arista = $_GET["cod_arista"];
    $sql = "select * from arista where cod_arista=".$_cod_arista;
    $result=$conn->query($sql);
    if(count($result)>0){
        $row = $result->fetch_assoc();
        $cod_arista = $row["cod_arista"];
        $cod_nodo_1 = $row["cod_nodo_1"];
        $cod_nodo_2 = $row["cod_nodo_2"];
    }
}else if($_POST){
    if (! empty($_POST["cod_arista"]) || $_POST["cod_arista"]==0){
        $cod_arista = $_POST["cod_arista"];
        $cod_nodo_1 = $_POST["txtnodoi"];
        $cod_nodo_2 = $_POST["txtnodof"];
        $sql = "";

        if($cod_arista > 0){
            $sql = "update arista set cod_nodo_1='".$_cod_nodo_1."', cod_nodo_2='".$_cod_nodo_2.
                "' where cod_arista=".$_cod_arista;

        }else{
            $sql = "insert into arista (cod_nodo_1, cod_nodo_2, estado) values ";
            $sql .= "('".$_cod_nodo_1."', '".$_cod_nodo_2."',1 )";
            echo '<script language="javascript">alert("'.$sql.'");</script>';
            $sql = "insert into arista (cod_nodo_1, cod_nodo_2, estado) values ";
            $sql .= "('".$_cod_nodo_2."', '".$_cod_nodo_1."',1 )";
            echo '<script language="javascript">alert("'.$sql.'");</script>';
        }
        $res=$conn->query($sql);
        if ($res == 1 ) {
            header("location: arista.php");
        }
    }
}

$sqlnodos1 = "select * from nodo where estado = 1";
$nodo1=$conn->query($sqlnodos1);
$sqlnodos2 = "select * from nodo where estado = 1";
$nodo2=$conn->query($sqlnodos2);
?>

```

c) Script “destino.php”

```

<?php
include("../system/src/conexion.php");
if (! empty($_GET["cod_destino"])){
    $cod_destino = $_GET["cod_destino"];
    if($cod_destino > 0){
        $sql = "update destino set estado = 0 where cod_destino=".$_cod_destino;
        $res=$conn->query($sql);
        if ($res == 1 ) {
            header("location: destino.php");
        }
    }
}

$sql = " select d.cod_destino, d.d_nombre, d.descripcion, d.cod_usuario, d.cod_nodo,
    d.imagen, d.estado, u.u_nombre, n.n_nombre FROM destino d INNER JOIN usuario u
    ON d.cod_usuario = u.cod_usuario INNER JOIN nodo n ON d.cod_nodo = n.cod_nodo
    where d.estado = 1";
$result=$conn->query($sql);
?>

```

d) Script “destino_frm.php”

```

<?php
include("../system/src/conexion.php");
$res="";
$cod_destino = 0;

```

```

$cod_nodo = 0;
$d_nombre= '';
$descripcion='';
$imagen='';
$estado='';

if (! empty($_GET["cod_destino"])){
    $cod_destino = $_GET["cod_destino"];
    $sql = "select * from destino where cod_destino=".$_cod_destino;
    $result=$conn->query($sql);
    if(count($result)>0){
        $row = $result->fetch_assoc();
        $cod_destino = $row["cod_destino"];
        $cod_nodo = $row["cod_nodo"];
        $d_nombre = $row["d_nombre"];
        $descripcion= $row["descripcion"];
        $imagen = $row["imagen"];
        $estado= $row["estado"];
    }
}else if($_POST){
    if (! empty($_POST["cod_destino"]) || $_POST["cod_destino"]==0){
        $cod_destino = $row["cod_destino"];
        $cod_nodo = $row["cod_nodo"];
        $d_nombre = $row["d_nombre"];
        $descripcion= $row["descripcion"];
        $imagen = $row["imagen"];
        $estado= $row["estado"];
        $nom=$_REQUEST["txtnom"];
        $foto=$_FILES["foto"]["name"];
        $ruta=$_FILES["foto"]["tmp_name"];
        $destino="fotos/".$foto;
        copy($ruta,$destino);

        $sql = "";
        if($cod_destino > 0){
            $sql = "update destino set d_nombre='".$_d_nombre."', descripcion='".$_descripcion."',
                cod_nodo = '".$_cod_nodo."', imagen = '".$_imagen.'" where cod_destino=".$_
                $cod_destino;
        }else{
            $sql = "insert into destino (d_nombre, descripcion, cod_nodo, cod_usuario, imagen,
                estado) values ";
            $sql .= " ('".$_d_nombre."', '".$_descripcion."', '".$_cod_nodo."', '".$_SESSION
                ["cod_usuario"] ."', '".$_imagen."',1 )";
            echo "<script language='javascript'>alert('".$_sql."');</script>";
        }

        $res=$conn->query($sql);
        if ($res == 1 ) {
            header("location: destino.php");
        }
    }
}

$sqlnodos = "select * from nodo where estado = 1";
$nodo=$conn->query($sqlnodos);
?>

```

e) Script “nodo.php”

```

<?php include("../system/src/conexion.php");
if (! empty($_GET["cod_nodo"])){
    $cod_nodo = $_GET["cod_nodo"];
    if($cod_nodo > 0){
        $sql = "update nodo set estado = 0 where cod_nodo=".$_cod_nodo;
        $res=$conn->query($sql);
        if ($res == 1 ) {
            header("location: nodo.php");
        }
    }
}
}

```

```

$sql = " select n.cod_nodo, n.n_nombre , n.coordenada_x, n.coordenada_y, n.estado, u.u_nombre
        from nodo n INNER JOIN usuario u on n.cod_usuario = u.cod_usuario where n.cod_nodo =
        cod_nodo and n.estado = 1";
$result=$conn->query($sql);
?>

```

f) Script “nodo_frm.php”

```

<?php include("../system/src/conexion.php");
$res="";
$cod_nodo = 0;
$cod_usuario = 1;
$n_nombre='';
$coordenada_x='';
$coordenada_y='';
if (! empty($_GET["cod_nodo"])){
    $cod_nodo = $_GET["cod_nodo"];
    $sql = "select * from nodo where cod_nodo=".$_cod_nodo;
    $result=$conn->query($sql);
    if(count($result)>0){
        $row = $result->fetch_assoc();
        $cod_nodo = $row["cod_nodo"];
        $n_nombre = $row["n_nombre"];
        $coordenada_x= $row["coordenada_x"];
        $coordenada_y= $row["coordenada_y"];
        $cod_usuario = $row["cod_usuario"];
    }
}else if($_POST){
    if (! empty($_POST["id"]) || $_POST["id"]==0){
        $cod_nodo = $_POST["id"];
        $n_nombre = $_POST["txtn_nombre"];
        $coordenada_x = $_POST["txtcoordenada_x"];
        $coordenada_y = $_POST["txtcoordenada_y"];
        $cod_usuario = $row["txtcod_usuario"];
        $sql = "";
        if($cod_nodo > 0){
            $sql = "update nodo set coordenada_x = '". $coordenada_x ."', coordenada_y =
                    '". $coordenada_y ."', n_nombre = '". $n_nombre ."' where cod_nodo=".$_cod_nodo;
        }else{
            $sql = "insert into nodo (n_nombre,coordenada_x,coordenada_y,cod_usuario,estado)
                    values ";
            $sql .= "(''. $n_nombre .'', '". $coordenada_x .'', '". $coordenada_y .'', "'. $_SESSION
                    ["cod_usuario"] .",1)";
            echo '<script language="javascript">alert(''. $sql .'');</script>';
        }
        $res=$conn->query($sql);
        if ($res == 1 ) {
            header("location: nodo.php");
            //echo '<script language="javascript">alert(''. $sql .'');</script>';
        }
    }
}
?>

```

g) Script “usuario.php”

```

<?php include("../system/src/conexion.php");
if (! empty($_GET["cod_usuario"])){
    $cod_usuario = $_GET["cod_usuario"];
    if($cod_usuario > 0){
        $sql = "update usuario set estado = 0 where cod_usuario=".$_cod_usuario;
        $res=$conn->query($sql);
        if ($res == 1 ) {
            header("location: usuario.php");
        }
    }
}
$sql = "select * from usuario where estado = 1 and tipo = 0";
$result=$conn->query($sql);

```

?>

h) Script “usuario_frm.php”

```
<?php include("../system/src/conexion.php");

$res="";
$cod_usuario = 0;
$nombres="";
$ap_pat = '';
$ap_mat = '';
$correo = '';
$usuario = '';
$contrasena = '';
$tipo= 0;
if (! empty($_GET["cod_usuario"])) {
    $id = $_GET["cod_usuario"];
    $sql = "select * from usuario where cod_usuario=".$id." and estado = 1 ";
    $result=$conn->query($sql);
    if(is_integer(count($result))>0){
        $row = $result->fetch_assoc();
        $cod_usuario = $row["cod_usuario"];
        $nombres = $row["u_nombre"];
        $ap_pat = $row["ap_pat"];
        $ap_mat = $row["ap_mat"];
        $correo = $row["correo"];
        $usuario = $row["usuario"];
        $contrasena = $row["contrasena"];
        $tipo = $row["tipo"];
        echo '<script language="javascript">console.log("llena usuario");</script>';
    }
}
else if($_POST){
    if (! empty($_POST["id"]) || $_POST["id"]==0){
        $cod_usuario= $_POST["id"];
        $nombres= $_POST["txtNombres"];
        $ap_pat= $_POST["txtAp_pat"];
        $ap_mat= $_POST["txtAp_mat"];
        $correo= $_POST["txtcorreo"];
        $usuario = $_POST["txtUsuario"];
        $contrasena= $_POST["txtClave"];
        $tipo= $_POST["txtTipo"];
        $sql = "";
        if($cod_usuario > 0){
            $sql = "update usuario set u_nombre='".$nombres."', ap_pat='".$ap_pat."', ap_mat='".$ap_mat."', correo='".$correo."', usuario='".$usuario."', contrasena='".$contrasena."' where cod_usuario=".$cod_usuario;
            echo '<script language="javascript">console.log("actualiza");</script>';
        }
        else{
            $sql = "insert into usuario(u_nombre,ap_pat, ap_mat,correo,usuario, contrasena, tipo,estado) values ";
            $sql .= "('" . $nombres . "', '" . $ap_pat . "', '" . $ap_mat . "', '" . $correo . "', '" . $usuario . "', '" . $contrasena . "',0,1)";
            echo '<script language="javascript">console.log("'" . $sql . "');</script>';
        }
        echo '<script language="javascript">console.log("'" . $sql . "');</script>';
        $res=$conn->query($sql);
        if ($res == 1 ) {
            header("location: usuario.php");
        }
    }
}
?>
```

i) Script “ValidarUsuario.php”

```
<?php
include("../system/src/conexion.php");
$susu =$_POST["txtUsuario"];
$pass =$_POST["txtClave"];
```



```

$sql="select * from usuario where usuario='$usu' and contrasena = '$pass'";
$result=$conn->query($sql);

if($result->num_rows > 0){ //si hay resultados se validara
    while($row = $result->fetch_assoc()){
        session_start();
        $_SESSION["cod_usuario"] = $row["cod_usuario"];
        $_SESSION["u_nombre"] = $row["u_nombre"];
        $_SESSION["ap_pat"] = $row["ap_pat"];
        $_SESSION["ap_mat"] = $row["ap_mat"];
        $_SESSION["tipo"] = $row["tipo"];
    }
    header("location: ../inicio.php");
}else{
    header("location: ../index.php");
}
$conn->close();
?>

```

j) Script “ValidarLogin.php”

```

<?php
session_start();
if(empty($_SESSION["cod_usuario"])){
    header("location: index.php");
}
?>

```

3) Paquete Controlador

a) Script “errores.php”

```

<?
Class Errores extends Controller{
    function _construct(){
        parent::_construct();
        $this->view->mensaje =“Error genérico”;
        $this->view->render('errores/index');
    }
}
?

```

b) Script “controlador.php”

```

<?php
Class Controlador{
    //Cargar modelos
    Public function modelo($modelo){
        //Cargar
        Require_once '../modelo/'.$modelo.'.php'
        //Instancias el modelo
        Return new $modelo();
    }
    //cargar vista
    Public function modelo($modelo){
        //verificar la existencia de la vista
        If(file_exists('../vista'.$vista.'.php'))
        Requiere_once '../vista'.$vista.'.php'
        Else{
            Die('la vista no existe')
        }
    }
}
?>

```

4) Paquete “System”

a) Script “Style.css”

```
.text-right{text-align: right!important;}
.text-left{text-align: left!important;}
.text-center{text-align: center!important;}

body{
background:#E0F2F1; // #E0F2F1
}
.login{
background:#EDC63D;
}
#lienzo{
background: url(..img/MapaUNPRG.jpg);
}

#contenedor{
position:relative;
margin:5vh auto;
width:4200px;
height:2700px;
}
```

b) Script “Conexión.php”

```
<?php
$servername="127.0.0.1";
$username="root";
$password="root";
$dbname="bd_tesis";
echo '<script language="javascript">console.log("entro");</script>';
//crear conexion
$conn= new mysqli($servername, $username, $password, $dbname);

//verificar conexion
if($conn->connect_error){
    die("Connection failed: " . $conn->connect_error);
    echo '<script language="javascript">alert("Error");</script>';
}

?>
```

c) Paquete “Img”



Figura 60: Login.jpg
Fuente: Elaboración Propia

ANEXO 7: Interfaz de Programación de Aplicaciones

1) Script “.htaccess”

```
Options +FollowSymLinks
RewriteEngine On
RewriteRule ^([a-zA-Z0-9]*)$ datos.php?variable=$1
```

2) Script “conexión.php”

```
<?php
class DB{
    private $host;
    private $db;
    private $user;
    private $password;
    private $charset;
    public function __construct(){
        $this->host      = '127.0.0.1';
        $this->db        = 'bd_tesis';
        $this->user       = 'root';
        $this->password   = "root";
        //$this->password = "6#vWHD_$";
        $this->charset    = 'utf8mb4';
    }

    function connect(){

        try{
            $connection = "mysql:host=".$this->host.";dbname=" . $this->db . ";charset= "
                . $this->charset;
            $options = [
                PDO::ATTR_ERRMODE            => PDO::ERRMODE_EXCEPTION,
                PDO::ATTR_EMULATE_PREPARES   => false,
            ];
            $pdo = new PDO($connection,$this->user,$this->password);
            return $pdo;

        }catch(PDOException $e){
            print_r('Error connection: ' . $e->getMessage());
        }
    }
}
?>
```

3) Script “consultas.php”

```
<?php
include_once 'conexion.php';
class Nodo extends DB{

    function obtenerNodo(){
        $query = $this->connect()->query('SELECT * FROM nodo');
        return $query;
    }
}

class Arista extends DB{

    function obtenerArista(){
        $query = $this->connect()->query('SELECT * FROM arista');
        return $query;
    }
}

class Destino extends DB{
```

```

function obtenerDestino(){
    $query = $this->connect()->query('SELECT * FROM destino');
    return $query;
}
}
?>

```

4) Script “datos.php”

```

<?php
header('Content-Type: application/json');
header("Access-Control-Allow-Origin: *");
include_once 'consultas.php';

if($_GET['variable']== 'nodo'){
    $nodo = new Nodo();
    $nodos = array();
    $nodos["nodos"] = array();
    $res = $nodo->obtenerNodo();
    if($res->rowCount()){
        while ($row = $res->fetch(PDO::FETCH_ASSOC)){

            $item=array(
                'cod_nodo'=> $row['cod_nodo'],
                'coordenada_x'=> $row['coordenada_x'],
                'coordenada_y'=> $row['coordenada_y'],
                'estado'=> $row['estado'],
            );
            array_push($nodos["nodos"], $item);
        }
        echo json_encode($nodos);
    }else{
        echo json_encode(array('mensaje' => 'No hay elementos'));
    }
}

}else{

    if($_GET['variable']== 'arista'){
        $arista = new Arista();
        $aristas = array();
        $aristas["aristas"] = array();
        $res = $arista->obtenerArista();

        if($res->rowCount()){

            while ($row = $res->fetch(PDO::FETCH_ASSOC)){

                $item=array(
                    'cod_arista'=> $row['cod_arista'],
                    'cod_nodo_1'=> $row['cod_nodo_1'],
                    'cod_nodo_2'=> $row['cod_nodo_2'],
                );
                array_push($aristas["aristas"], $item);
            }

            echo json_encode($aristas);
        }else{
            echo json_encode(array('mensaje' => 'No hay elementos'));
        }
    }

}

}else{
    if($_GET['variable']== 'destino'){
        $destino = new Destino();
        $destinos = array();
        $destinos["destinos"] = array();
        $res = $destino->obtenerDestino();
        if($res->rowCount()){
            while ($row = $res->fetch(PDO::FETCH_ASSOC)){

```

```

        $item=array(
            'cod_destino'=> $row['cod_destino'],
            'nombre'=> $row['nombre'],
            'descripcion'=> $row['descripcion'],
            'cod_nodo'=> $row['cod_nodo'],
            'imagen'=> $row['imagen'],
            'estado'=> $row['estado'],
        );
        array_push($destinos["destinos"], $item);
    }

    echo json_encode($destinos);
}else{
    echo json_encode(array('mensaje' => 'No hay elementos'));
}
}else{
    echo 'solicitud no encontrada';
}
}
}
?>

```

ANEXO 8: Implementación de la aplicación móvil

1) Paquete de Datos

a) Script “API.cs”

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System.IO;

public class API : MonoBehaviour {

    public string URL;
    public GameObject[] ObjNodo;
    public GameObject[] ObjDestinos;
    public Text[] Titulos;
    public int Cantidad;
    public GameObject Algoritmo_A;

    public void Request(string Tabla)
    {

        WWW request = new WWW(URL+Tabla);

        if(Tabla=="nodo")
            StartCoroutine(NodoOnReponse(request));
        if(Tabla=="arista")
            StartCoroutine(AristaOnReponse(request));
        if(Tabla=="destino")
            StartCoroutine(DestinoOnReponse(request));
    }

    private IEnumerator NodoOnReponse(WWW req)
    {
        yield return req;
        ListaNodos listaNodos = JsonUtility.FromJson<ListaNodos> (req.text);
        Cantidad=listaNodos.nodos.Count;

        for (int i = 0; i < Cantidad; i++) {
            listaNodos.Ordenar (ObjNodo[i],i+1);
        }
    }
}

```

```

        this.Algoritmo_A.GetComponent<Busqueda_> ().CantidaNodos = Cantidad;
    }

    private IEnumerator AristaOnReponse(WWW req)
    {
        yield return req;
        ListaAristas listaAristas = JsonUtility.FromJson<ListaAristas> (req.text);

        for (int i = 0; i < Cantidad; i++) {
            listaAristas.Cargar (ObjNodo, ObjNodo[i], i+1);
        }
    }

    private IEnumerator DestinoOnReponse(WWW req)
    {
        int CantidadDestinos = 0;
        yield return req;
        ListaDestinos listaDestinos = JsonUtility.FromJson<ListaDestinos> (req.text);
        CantidadDestinos = listaDestinos.destinos.Count;

        for (int i = 0; i < CantidadDestinos; i++) {
            ObjDestinos [i].SetActive(true);
        }

        for (int j = 0; j < CantidadDestinos; j++) {
            listaDestinos.Cargar (Titulos[j], ObjNodo, ObjDestinos[j], j+1);
        }
    }
}

[System.Serializable]
public class Nodo {
    public string cod_nodo;
    public string coordenada_x;
    public string coordenada_y;
    public string estado;

    public override string ToString ()
    {
        return string.Format ("{0},{1},{2},{3}", cod_nodo, coordenada_x, coordenada_y, estado);
    }
}

[System.Serializable]
public class ListaNodos {
    public List<Nodo> nodos;

    public void Ordenar(GameObject nodoPos, int numero)
    {
        foreach (Nodo nodo in nodos) {
            if(int.Parse(nodo.cod_nodo)==numero)
                nodoPos.transform.position = new Vector3(float.Parse (nodo.coordenada_x), 0.5f, float.Parse
                    (nodo.coordenada_y));
        }
    }
}

[System.Serializable]
public class Arista {
    public string cod_arista;
    public string cod_nodo_1;
    public string cod_nodo_2;

    public override string ToString ()
    {
        return string.Format ("{0},{1},{2}", cod_arista, cod_nodo_1, cod_nodo_2);
    }
}

```

```

[System.Serializable]
public class ListaAristas {
    public List<Arista> aristas;

    public void Cargar (GameObject[] ObjNodo,GameObject nodoRel, int numero){
        int Cont = 0;
        int Cantidad = 0;

        foreach (Arista arista in aristas) {
            if (int.Parse (arista.cod_nodo_1) == numero) {
                Cantidad++;
            }
        }

        nodoRel.GetComponent<NodoPadre> ().Relac_Nodo=new GameObject [Cantidad];

        foreach (Arista arista in aristas) {
            if (int.Parse (arista.cod_nodo_1) == numero) {
                nodoRel.GetComponent<NodoPadre>().Relac_Nodo[Cont]=ObjNodo[int.Parse (arista.cod_nodo_2)
-1];
                Cont++;
            }
        }
    }
}

[System.Serializable]
public class Destino {
    public string cod_destino;
    public string nombre;
    public string descripcion;
    public string cod_nodo;
    public string imagen;
    public string estado;

    public override string ToString ()
    {
        Return string.Format ("{0},{1},{2},{3},{4},{5}",cod_destino,nombre,descripcion,cod_nodo,
imagen,estado);
    }
}

[System.Serializable]
public class ListaDestinos {
    public List<Destino> destinos;

    public void Cargar(Text titulo, GameObject[] ObjNodo,GameObject destinoPos,int numero){

        foreach (Destino destino in destinos) {
            if (int.Parse (destino.cod_destino) == numero) {
                destinoPos.GetComponent<DestinoBtn> ().Nodo_Destino=ObjNodo[int.Parse
(destino.cod_nodo)-1] ;
                destinoPos.GetComponent<DestinoBtn> ().Descripcion = destino.descripcion;
                destinoPos.GetComponent<DestinoBtn> ().Link = destino.imagen;
                destinoPos.GetComponent<DestinoBtn> ().Nombre = destino.nombre;
                titulo.text = destino.nombre;
            }
        }
    }
}

```

b) Script “DestinoBtn.cs”

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class DestinoBtn : MonoBehaviour {

    public GameObject Algoritmo_A;
    public InputField Input_Nodo_Destino;
    public InputField Input_Nodo_Descripcion;
    public GameObject Cargar_Imagen;

    public GameObject Nodo_Destino;
    public string Nombre;
    public string Descripcion;
    public string Link;

    public void Enviar_Datos_Destino(){
        this.Algoritmo_A.GetComponent<Busqueda_> ().Nodo_Final = Nodo_Destino;
        Input_Nodo_Destino.text = Nombre;
        Input_Nodo_Descripcion.text = Descripcion;
        Cargar_Imagen.GetComponent<CargarImagen> ().url = Link;
    }
}
```

2) Paquete Lógico

a) Script “Activar Búsqueda.cs”

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Activar_Busqueda : MonoBehaviour {
    public GameObject Btn_Azul;
    public GameObject Btn_Rojo;
    public InputField Input_Destino;
    public InputField Input_Origen;

    void Update ()
    {
        if (Input_Destino.text == ""||Input_Origen.text == "") {
            Btn_Azul.SetActive(false);
            Btn_Rojo.SetActive(true);
        } else {
            if (Input_Destino.text == Input_Origen.text) {
                Btn_Azul.SetActive(false);
                Btn_Rojo.SetActive(true);
            } else {
                Btn_Azul.SetActive(true);
                Btn_Rojo.SetActive(false);
            }
        }
    }
}
```

b) Script “Algoritmo A Star.cs”

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```



```

public class Buscador : MonoBehaviour {
    public bool activar;
    public List<NodoEspecial> caminoTotal;
    public List<NodoEspecial> caminoFinal;
    public GameObject inicio;
    public GameObject final;
    public int pasoActual;
    private LineRenderer Linea;

    void Start () {
        this.activar = false;
        this.pasoActual = 0;
        Linea = GetComponent<LineRenderer> ();
    }

    void Update () {
        if(this.activar){
            this.caminoTotal = new List<NodoEspecial>();
            this.caminoFinal = new List<NodoEspecial>();
            this.pasoActual = 0;
            AlgoritmoEstrella();
            this.activar = false;
        }
    }

    public void AlgoritmoEstrella(){
        NodoEspecial nuevo = new NodoEspecial(this.inicio, 0,
        false, 99999);
        this.caminoTotal.Add(nuevo);
        Enjambre();
        for(int i = 0; i < this.caminoTotal.Count; i++){
            if(this.caminoTotal[i].actual){
                this.caminoFinal.Add(this.caminoTotal[i]);
            }
        }
        Linea.positionCount = this.caminoFinal.Count;
        for(int i = 0; i < this.caminoFinal.Count; i++){
            Linea.SetPosition (i, this.caminoFinal
            [i].nodo.transform.position);
        }
    }

    public void Enjambre(){
        float minor = this.caminoTotal[0].heuris;
        int location = 0;
        for(int i = 0; i < this.caminoTotal.Count; i++){
            if(this.caminoTotal[i].heuris <= minor){
                minor = this.caminoTotal[i].heuris;
                location = i;
            }
        }
        if(this.caminoTotal[location].nodo == this.final){
            this.caminoTotal[location].actual = true;
            Debug.Log("Done");
            return;
        }
        if(this.caminoTotal.Count > 999){
            Debug.Log("Tope de emergencia");
            return;
        }
        Evaluar(location);
        Enjambre();
    }

    public void Evaluar(int locate){
        if(this.caminoTotal[locate].paso == this.pasoActual){
        }else{
            float wminor = 99999;
            for(int i = 0; i < this.caminoTotal.Count; i++){
                if(this.caminoTotal[i].paso == this.pasoActual){

```

```

        if(this.caminoTotal[i].heuris < wminor){
            wminor = this.caminoTotal[i].heuris;
        }
    }
    this.pasoActual = this.caminoTotal[locate].paso;
    while(this.caminoTotal[this.caminoTotal.Count-1].paso >
this.caminoTotal[locate].paso){
        this.caminoTotal.RemoveAt(this.caminoTotal.Count-1);
    }
    for(int i = 0; i < this.caminoTotal.Count; i++){
        if(this.caminoTotal[i].paso == this.pasoActual){
            if(this.caminoTotal[i].actual){
                this.caminoTotal[i].heuris = wminor;
                this.caminoTotal[i].actual = false;
            }
        }
    }
    Debug.Log("Regresar");
}
this.caminoTotal[locate].actual = true;
this.pasoActual += 1;
AbrirNodo(locate);
}

public void AbrirNodo(int locate){
    for(int i = 0; i < this.caminoTotal[locate]
.nodo.GetComponent<NodoV2>().NodeCollection.Length; i++){
        NodoEspecial extra = newNodoEspecial(this.caminoTotal[locate].nodo. GetComponent<NodoV2>().
NodeCollection[i],this.pasoActual,false,MedirHeuristica(this.caminoTotal[locate].
nodo,this.caminoTotal[locate].no do.GetComponent<NodoV2>().NodeCollection[i]));
        this.caminoTotal.Add(extra);
    }
}

public float MedirHeuristica(GameObject ini, GameObject fin){
    float res = 0;
    res += Vector3.Distance(ini.transform.position, fin.transform.position);
    res += Vector3.Distance(fin.transform.position, this.final.transform.position);
    return res;
}

[System.Serializable]
public class NodoEspecial{
    public GameObject nodo;
    public int paso;
    public bool actual;
    public float heuris;
    public NodoEspecial(GameObject nodo, int paso, bool actual, float heuris){
        this.nodo = nodo;
        this.paso = paso;
        this.actual = actual;
        this.heuris = heuris;
    }
}

```

c) Script “Geolocalización.cs”

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Geolocalizacion : MonoBehaviour {

    public float latitude;
    public float longitude;

    public void EnviarDatos(){

```

```

        StartCoroutine(localizar());
    }

    public IEnumerator localizar() {
        if (!Input.location.isEnabledByUser){
            Debug.Log("User has not enabled GPS");
            yield break;
        }
        Input.location.Start ();
        int maxWait = 20;
        while (Input.location.status == LocationServiceStatus.Initializing && maxWait> 0){
            yield return new WaitForSeconds(1);
            maxWait--;
        }
        if (maxWait <1){
            Debug.Log("timed out");
            yield break;
        }
        if (Input.location.status == LocationServiceStatus.Failed ){
            Debug.Log ("No se puede determinar la ubicación del dispositivo");
            yield break;
        } else{
            latitude=Input.location.lastData.latitude;
            longitude=Input.location.lastData.longitude;
        }
    }
}

```

d) Script “Hide Canvas Up.cs”

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class HideCanvasUp : MonoBehaviour {

    private Vector3 alturaInicial;
    private bool upMove;
    private bool activate;
    public float velocidad;

    void Start () {
        this.upMove = false;
        this.alturaInicial = this.transform.position;
    }

    void Update () {
        if (this.activate) {
            if (this.upMove) {
                this.transform.position -= new Vector3 (this.velocidad, 0,0);
                if (this.alturaInicial.x - this.GetComponent<RectTransform> ().rect.width*2 >
                    this.transform.position.x) {
                    this.activate = false;
                    Vector3 extra = this.alturaInicial - new Vector3 (this.GetComponent<RectTransform>
                        ().rect.width*2, 0,0);
                    this.transform.position = extra;
                }
            } else {
                this.transform.position += new Vector3 (this.velocidad, 0,0);
                if (this.transform.position.x > this.alturaInicial.x) {
                    this.activate = false;
                    Vector3 extra = this.alturaInicial;
                    this.transform.position = extra;
                }
            }
        }
    }
}

```

```

public void Activar(){
    this.upMove = !this.upMove;
    this.activate = true;
}
}

```

e) Script “Movimiento Cámara.cs”

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Movimineto_Camara: MonoBehaviour {

    Vector2 touchDeltaPosition;
    float Velocidad_x;
    float Velocidad_y;

    void Update () {
        if(Input.touchCount==1)
        {
            Touch touchZero = Input.GetTouch(0);
            if(touchZero.phase == TouchPhase.Moved)
            {
                touchDeltaPosition= Input.GetTouch(0).deltaPosition;
                if (transform.position.x >= 260 && touchDeltaPosition.x < 0 || transform.position.x <= -
                    245 && touchDeltaPosition.x > 0 )
                    Velocidad_x=0;
                else
                    Velocidad_x=0.14f;
                if (transform.position.z >= 245 && touchDeltaPosition.y < 0 || transform.position.z <=
                    -65 && touchDeltaPosition.y > 0 )
                    Velocidad_y=0;
                else
                    Velocidad_y=0.14f;
                transform.Translate(-touchDeltaPosition.x * Velocidad_x,-touchDeltaPosition.y *
                    Velocidad_y,0);
            }
        }
    }
}

```

f) Script “Zoom.cs”

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Zoom : MonoBehaviour {

    public float perspectiveZoomSpeed = 0.5f;
    public float orthoZoomSpeed = 0.5f;
    public Camera primaryCam;

    void Update () {
        if (Input.touchCount == 2)
        {
            Touch touchZero = Input.GetTouch(0);
            Touch touchOne = Input.GetTouch(1);
            Vector2 touchZeroPrevPos = touchZero.position - touchZero.deltaPosition;
            Vector2 touchOnePrevPos = touchOne.position - touchOne.deltaPosition;
            float prevTouchDeltaMag = (touchZeroPrevPos - touchOnePrevPos).magnitude;
            float touchDeltaMag = (touchZero.position - touchOne.position).magnitude;
            float deltaMagnitudeDiff = prevTouchDeltaMag - touchDeltaMag;
            primaryCam.fieldOfView += deltaMagnitudeDiff * perspectiveZoomSpeed;
            primaryCam.fieldOfView = Mathf.Clamp(primaryCam.fieldOfView, 25.1f, 50.1f);
        }
    }
}

```

g) Script “Union.cs”

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[ExecuteInEditMode]
public class Union : MonoBehaviour {

    public GameObject Ubicacion;
    public GameObject NodoInicial;
    private LineRenderer Linea;
    private Transform point1;
    private Transform point3;
    public Vector3 point2;
    public LineRenderer lineRenderer;
    public int vertexCount =12;

    public void Conectar () {
        float Posx;
        float Posz;
        float DistanciaX;
        float DistanciaY;

        Posx=(NodoInicial.transform.position.x-Ubicacion.transform.position.x)/2+Ubicacion.
transform.position.x;
        Posz=(NodoInicial.transform.position.z-Ubicacion.transform.position.z)/2+Ubicacion.
transform.position.z;
        DistanciaX= NodoInicial.transform.position.x - Ubicacion.transform.position.x;
        DistanciaY= NodoInicial.transform.position.z - Ubicacion.transform.position.z;
        if(DistanciaX < 0)
            DistanciaX = DistanciaX * -1;
        if(DistanciaY < 0)
            DistanciaY = DistanciaY * -1;
        if(DistanciaX < DistanciaY){
            Posx=Posx+5f;
        }
        else{
            Posz=Posz+5f;
        }
        point1 = Ubicacion.transform;
        point2 = new Vector3(Posx,0.5f,Posz);
        point3 = NodoInicial.transform;
        var pointList= new List<Vector3>();
        for(float radio =0; radio <= 1; radio += 1.0f / vertexCount)
        {
            var tangentLineVertex1 = Vector3.Lerp(point1.position, point2, radio);
            var tangentLineVertex2 = Vector3.Lerp(point2, point3.position, radio);
            var bezierPoint = Vector3.Lerp(tangentLineVertex1, tangentLineVertex2, radio);
            pointList.Add(bezierPoint);
        }
        lineRenderer.positionCount = pointList.Count;
        lineRenderer.SetPositions(pointList.ToArray());
    }
}
```

h) Script “Nodo.cs”

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Nodo : MonoBehaviour {
    public GameObject[] Relac_Nodo;
    public float[] Fx;
    public int Estado=0;
    public int Expandido=0;
    public bool Activo = false;
}
```

i) Script “Detector.cs”

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Detector : MonoBehaviour {

    public double latitud_z;
    public double longitud_x;
    public GameObject Esfera;
    public GameObject NodeCollection;
    public GameObject NodoInicial;
    public GameObject A_Estrella;
    public GameObject Union;
    public GameObject GPS;
    public GameObject VentanaInfo;
    public InputField Input_Nodo_Org;
    public InputField Input_Ubicacion;

    public void Buscar(){

        float longitud;
        float latitud;
        latitud_z=GPS.GetComponent<Geolocalizacion> ().latitude;
        longitud_x=GPS.GetComponent<Geolocalizacion> ().longitude;

        longitud=(float) ((longitud_x+79.906730)*-100000);
        latitud=(float) ((latitud_z+6.706356)*-100000);

        Input_Ubicacion.text= "log: "+ longitud_x+ " lat: "+ latitud_z;
        if(longitud>-320 & longitud <350 & latitud>-100 & latitud <300){

            Esfera.transform.position= new Vector3(longitud,0.5f,latitud);
            Collider[] hitColliders = Physics.OverlapSphere(this.Esfera.transform.position, 10);

            if(hitColliders.Length!=0){

                NodeCollection=hitColliders[0].gameObject;
                this.A_Estrella.GetComponent<Busqueda_> ().Nodo_Inicial = NodeCollection;
                Input_Nodo_Org.text=NodeCollection.name;
                Debug.Log("nivel 1");
            }
            else{

                Collider[] hitColliders2 = Physics.OverlapSphere(this.Esfera.transform.position, 25);
                if(hitColliders2.Length!=0){

                    NodeCollection=hitColliders2[0].gameObject;
                    this.A_Estrella.GetComponent<Busqueda_> ().Nodo_Inicial = NodeCollection;
                    Input_Nodo_Org.text=NodeCollection.name;
                    Debug.Log("nivel 2");
                }
                else{

                    Collider[] hitColliders3 = Physics.OverlapSphere(this.Esfera.transform.position, 45);
                    if(hitColliders3.Length!=0){

                        NodeCollection=hitColliders3[0].gameObject;
                        this.A_Estrella.GetComponent<Busqueda_> ().Nodo_Inicial = NodeCollection;
                        Input_Nodo_Org.text=NodeCollection.name;
                        Debug.Log("nivel 3");
                    }
                    else{

                        Esfera.transform.position= new Vector3(-9.5f,0.5f,-2f);
                        NodeCollection=NodoInicial;
                    }
                }
            }
        }
    }
}
```

```

        Input_Nodo_Org.text=NodeCollection.name;
    }
}
else{
    Esfera.transform.position= new Vector3(-8.8f,0.5f,-10f);
    NodeCollection=NodoInicial;
    Input_Nodo_Org.text=NodeCollection.name;
    Input_Ubicacion.text= " Entrada Principal ";
    VentanaInfo.SetActive (true);
}

this.Union.GetComponent<Union> ().Ubicacion = Esfera;
this.Union.GetComponent<Union> ().NodoInicial = NodeCollection;
}
}

```

j) Script “Cargar Imagen.cs”

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class CargarImagen : MonoBehaviour {

    public RawImage Imagen;
    public Texture img;
    public string url;

    public void CargarImagenDestino() {
        StartCoroutine(LoadImageToUnity());
    }

    public IEnumerator LoadImageToUnity()
    {
        WWW W = new WWW(url);
        yield return W;
        Texture te = W.texture;
        img=te;
        Imagen.texture= te;
    }
}

```

k) Script “Load.cs”

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;

public class Load : MonoBehaviour {

    public Image BarraHorizontal;
    public float tiempo = 2f;
    private float ContadorTiempo;
    public Text Horizontal;
    private bool Bandera =false;

    void Start ()
    {
        ContadorTiempo = 0;
    }

    public void IniciarCargar()
    {
        Bandera =true;
    }
}

```

```

void Update ()
{
    if (Bandera == true) {

        if (ContadorTiempo <= tiempo) {
            ContadorTiempo = ContadorTiempo + Time.deltaTime;
            BarraHorizontal.fillAmount = ContadorTiempo / tiempo;
            Horizontal.text = (Convert.ToInt32 (100 * BarraHorizontal.fillAmount)).ToString () + "%";
        }else{
            Vector3 Pos = new Vector3 (-1753, 0,0);
            this.transform.position = Pos;
        }
    }
}
}

```

l) Script “Cargar Mapa.cs”

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class CargarMapa : MonoBehaviour {

    public Texture img;
    public GameObject mapa;
    public string url;

    public void CargarImagenDestino(){
        StartCoroutine(LoadImageToUnity());
    }

    public IEnumerator LoadImageToUnity()
    {
        WWW W = new WWW(url);
        yield return W;
        Texture te = W.texture;
        mapa.GetComponent<MeshRenderer>().material.mainTexture=te;
    }
}

```

m)Script “Exit.cs”

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Exit : MonoBehaviour {

    public void Salir(){
        Application.Quit()
    }
}

```

n) Script “Conexión www .cs”

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class ConexionWWW : MonoBehaviour {

    public Text Texto;
    public GameObject ObjBoton;

    void Start () {

```



```

        if(Application.internetReachability != NetworkReachability.NotReachable)
        {
            ObjBoton.SetActive(true);
        }else{
            Texto.text  ="Sin Conexion";
        }
    }
}

```

o) Script “Enfocar Ubicacion.cs”

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class EnfocarUbicacion : MonoBehaviour {

    public GameObject FlechaIndicadora;

    public void Reubicar () {
        Vector3 Pos = new Vector3(FlechaIndicadora.transform.position.x,150f,FlechaIndicadora.
            transform.position.z);
        this.transform.position = Pos;
    }
}

```

3) Materiales

a) Objeto 2D:

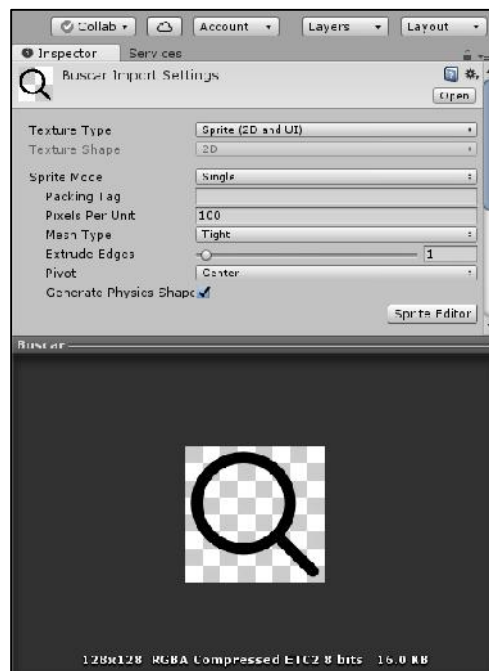


Figura 61: Sprite 2D Buscar
Fuente: Elaboración Propia

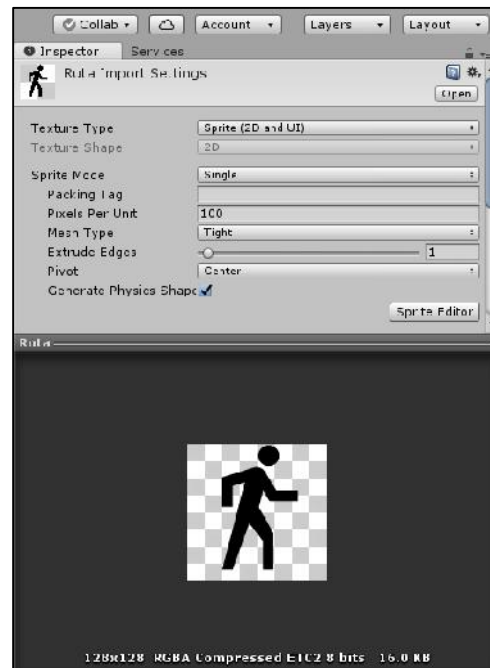


Figura 62: Sprite 2D Ruta
Fuente: Elaboración Propia

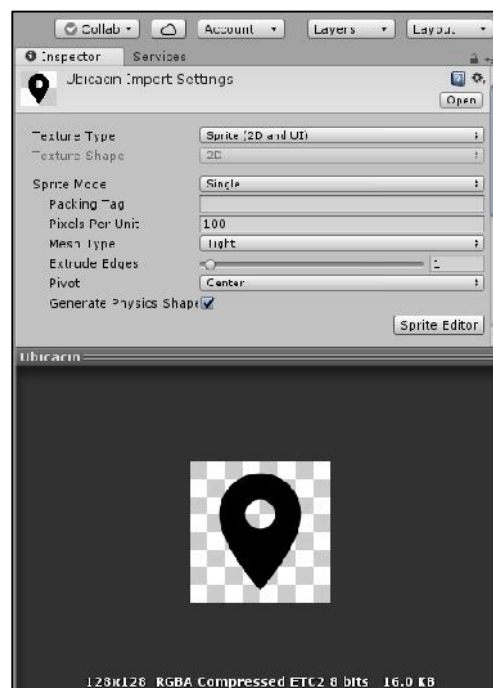


Figura 63: Sprite 2D Ubicacion
Fuente: Elaboración Propia

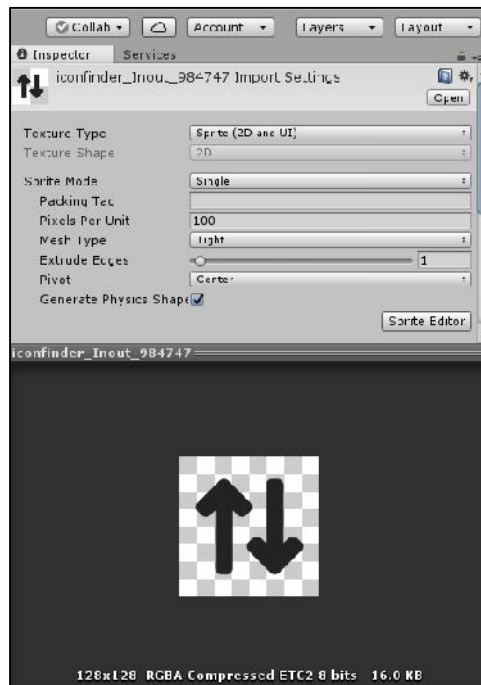


Figura 64: Sprite 2D Desplazar
Fuente: Elaboración Propia

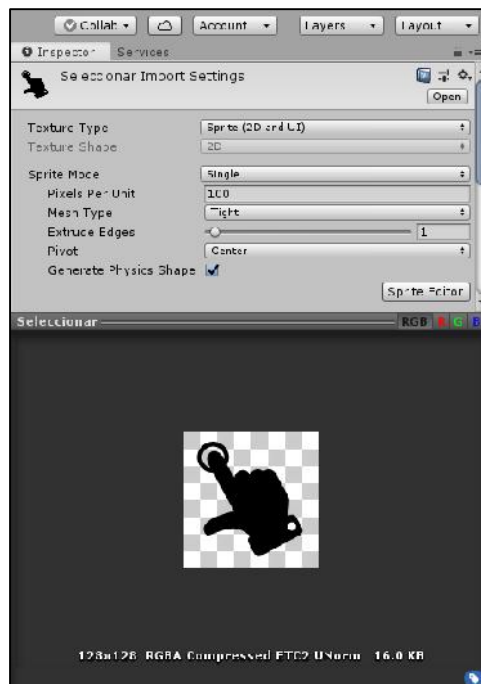


Figura 65: Sprite 2D Seleccionar
Fuente: Elaboración Propia

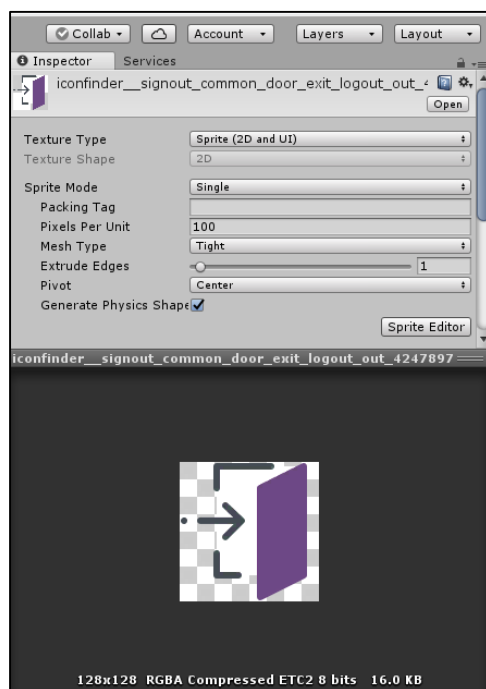


Figura 66: Sprite 2D Salir
Fuente: Elaboración Propia

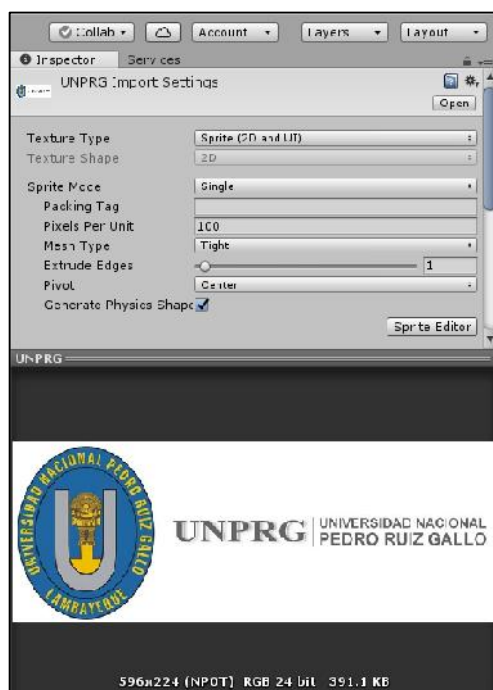


Figura 67: Sprite 2D UNPRG
Fuente: Elaboración Propia



Figura 68: Sprite 2D Icono
Fuente: Elaboración Propia

b) Objeto 3D:



Figura 69: Mesh de las aulas de Derecho, Objeto 3D
Fuente: Elaboración Propia

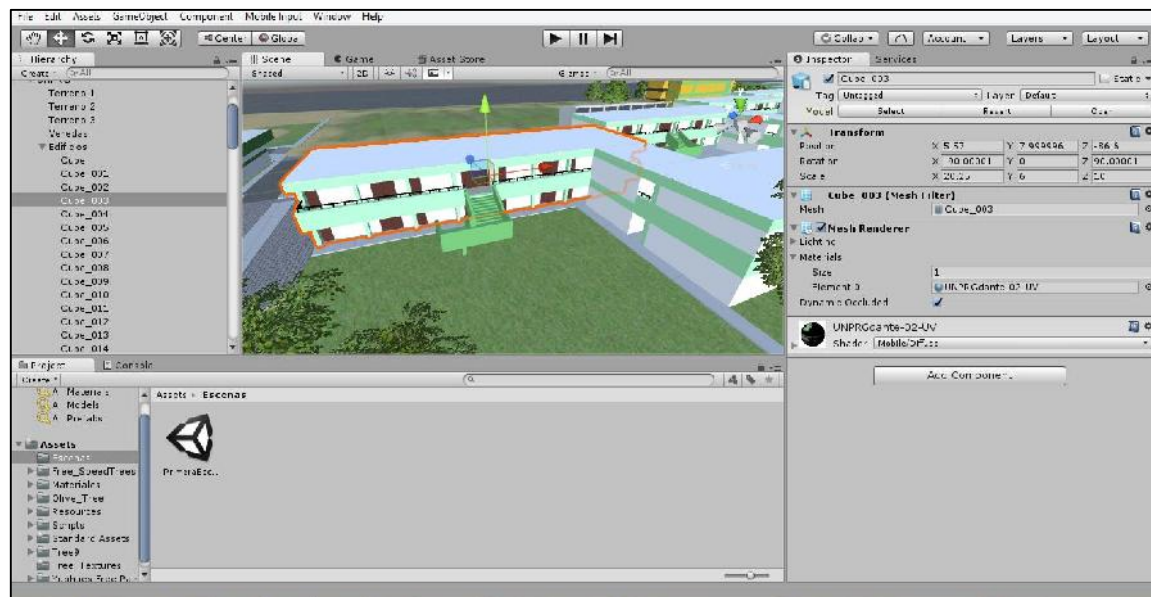


Figura 70: Mesh del Pabellón de Aulas, Objeto 3D
Fuente: Elaboración Propia



Figura 71: Mesh de las aulas block "A", Objeto 3D
Fuente: Elaboración Propia

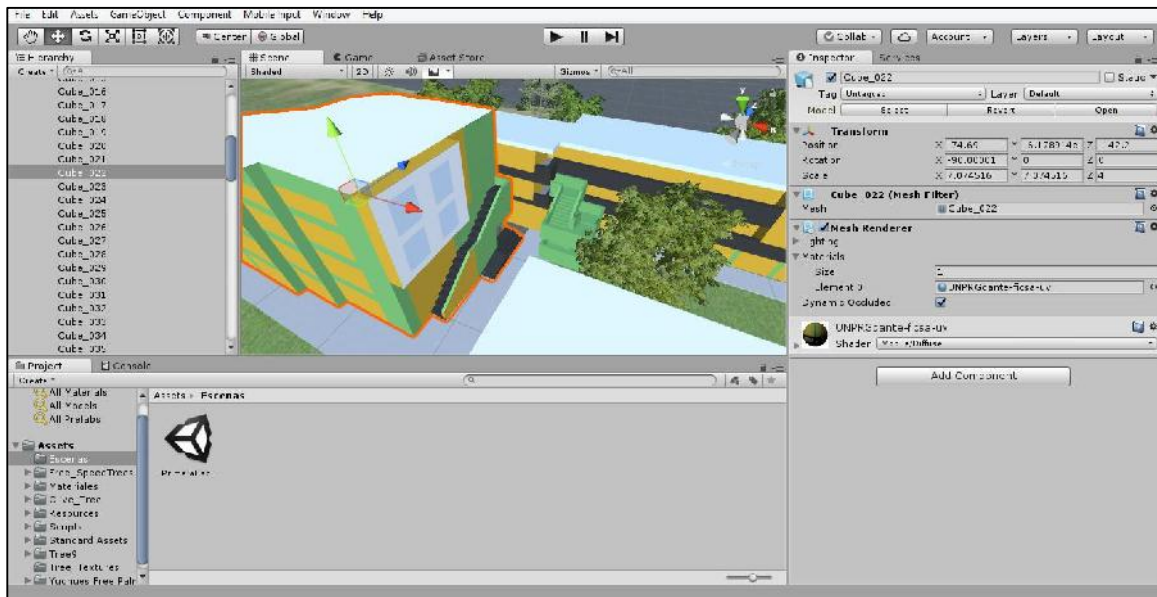


Figura 72: Mesh de FICSA, Objeto 3D
Fuente: Elaboración Propia

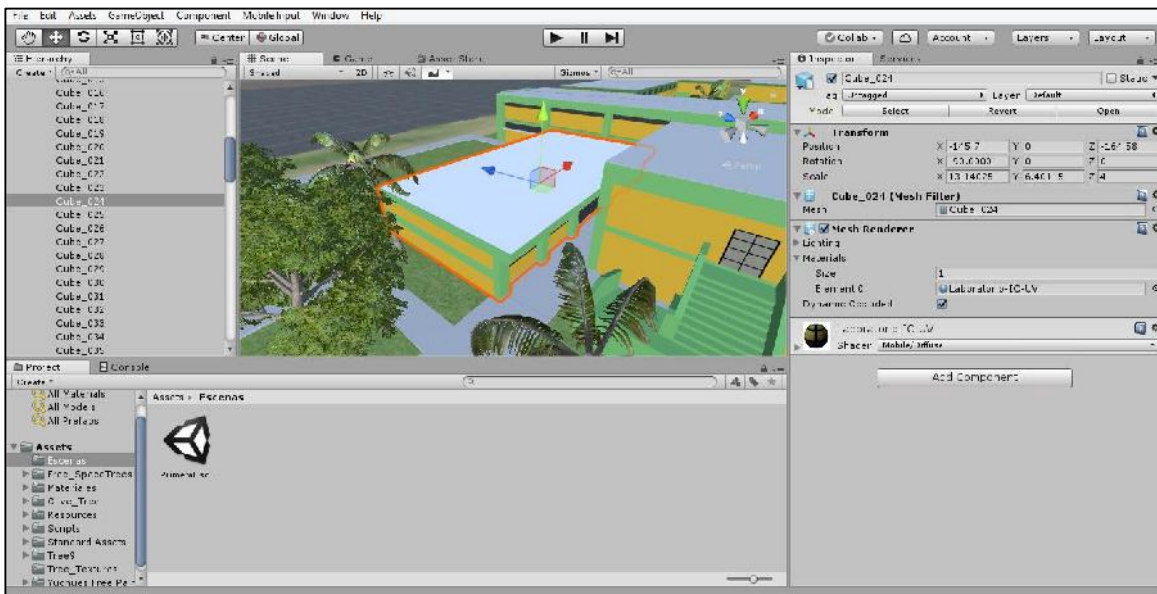


Figura 73: Mesh del laboratorio de ing. Civil, Objeto 3D
Fuente: Elaboración Propia

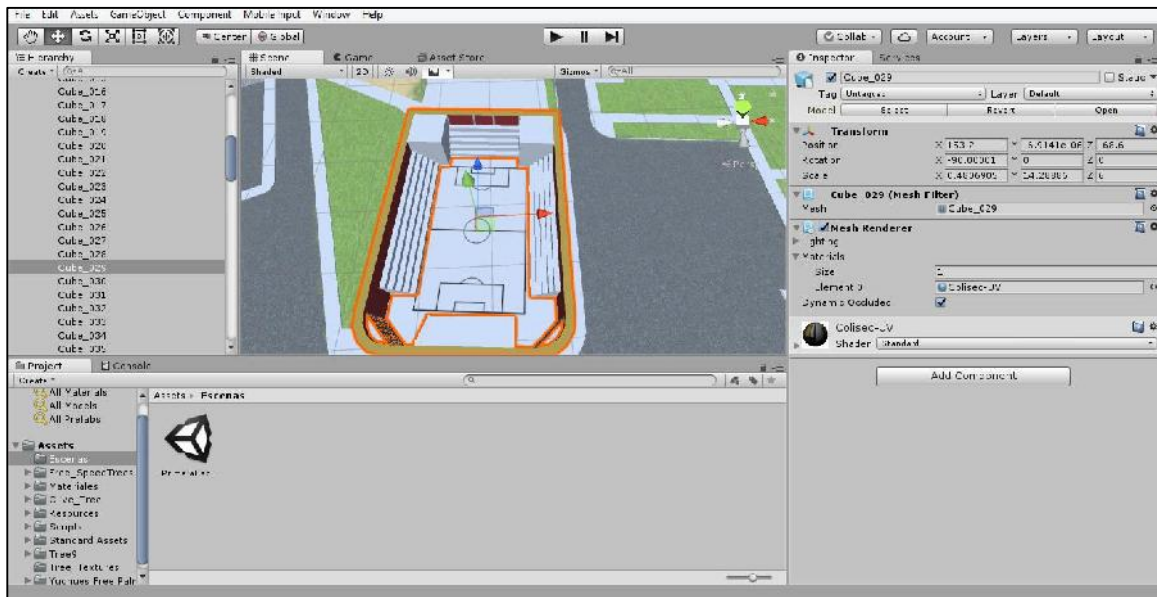


Figura 74: Mesh del Coliseo, Objeto 3D
Fuente: Elaboración Propia

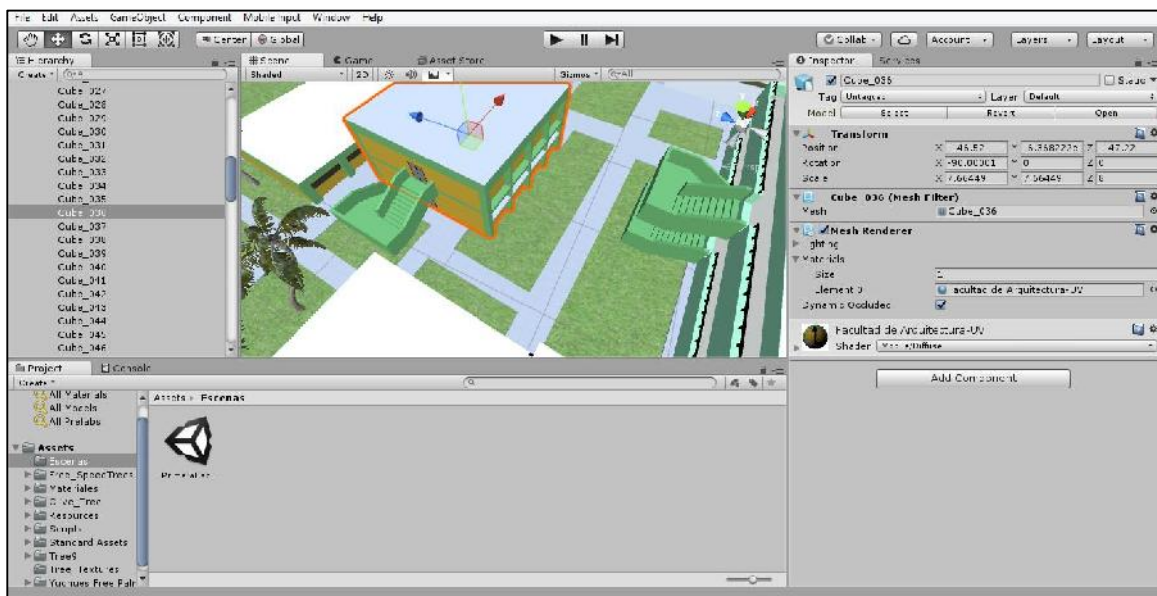


Figura 75: Mesh del Laboratorio de Arquitectura, Objeto 3D
Fuente: Elaboración Propia

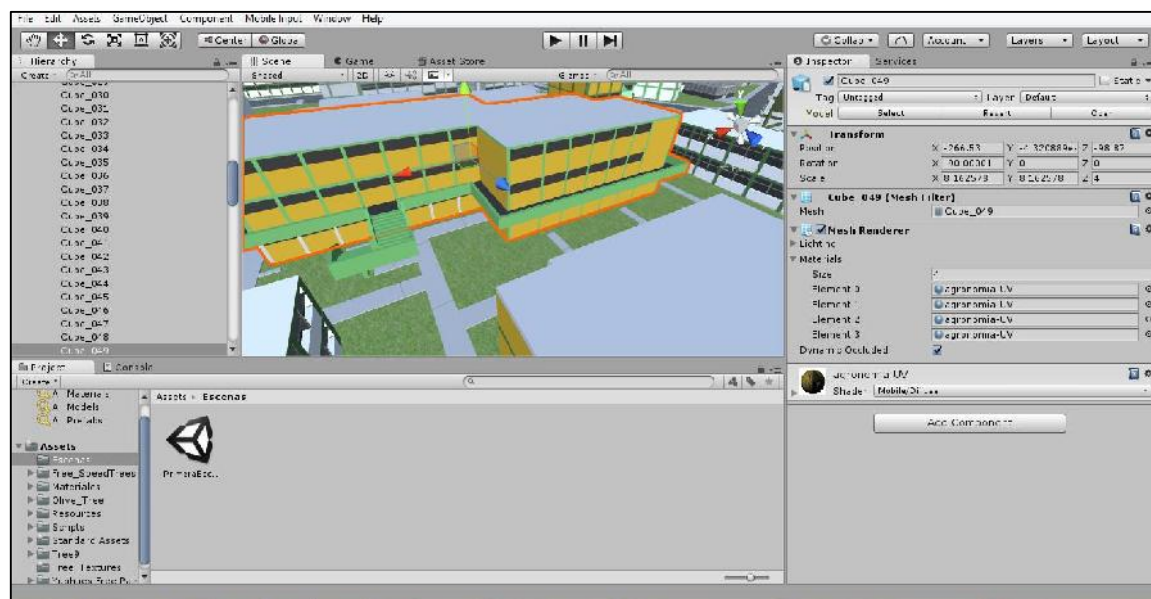


Figura 76: Mesh del laboratorio de Ciencias Biologicas, Objeto 3D
Fuente: Elaboración Propia

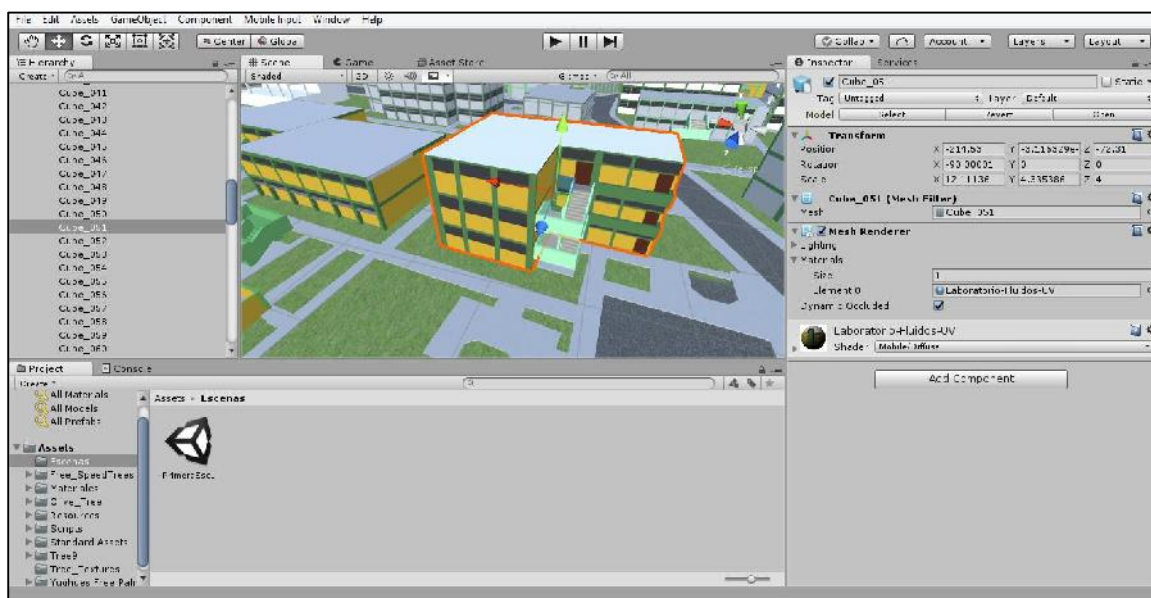


Figura 77: Mesh de Recursos Hidricos, Objeto 3D
Fuente: Elaboración Propia

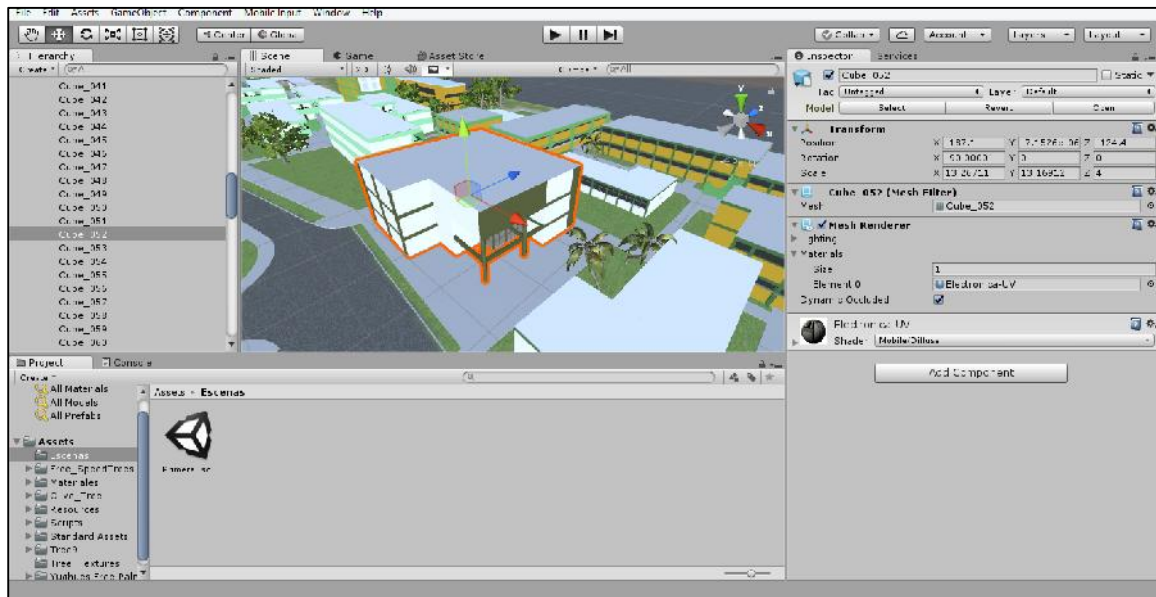


Figura 78: Mesh del Laboratorio de Electronica, Objeto 3D
Fuente: Elaboración Propia

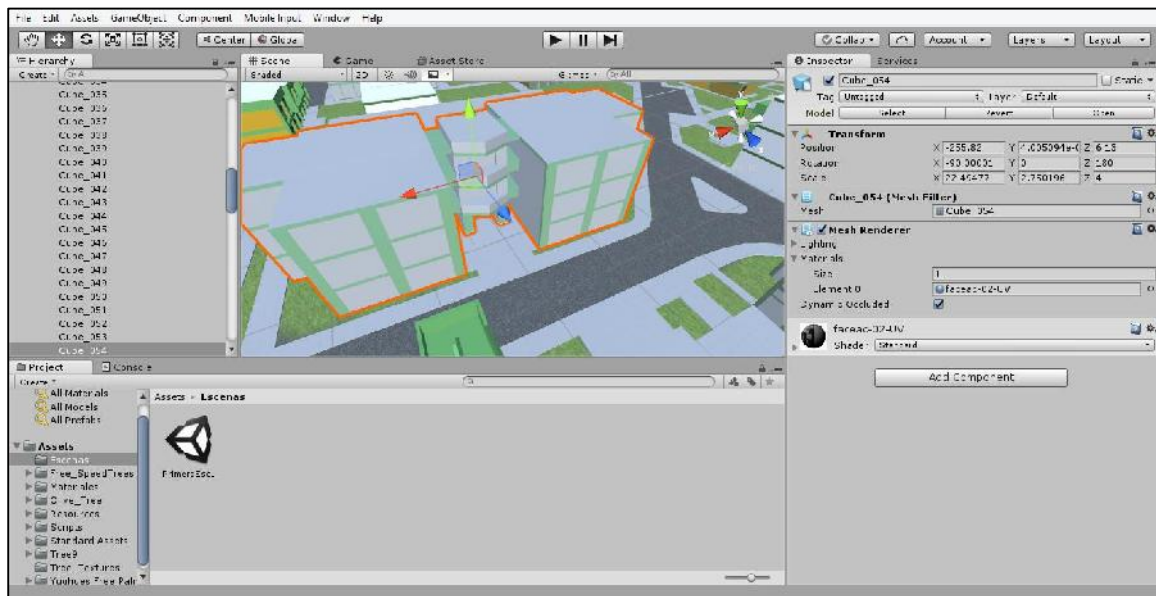


Figura 79: Mesh del pabello de aulas de FACEAC, Objeto 3D
Fuente: Elaboración Propia



Figura 80: Mesh de las aulas de Veterinaria, Objeto 3D
Fuente: Elaboración Propia

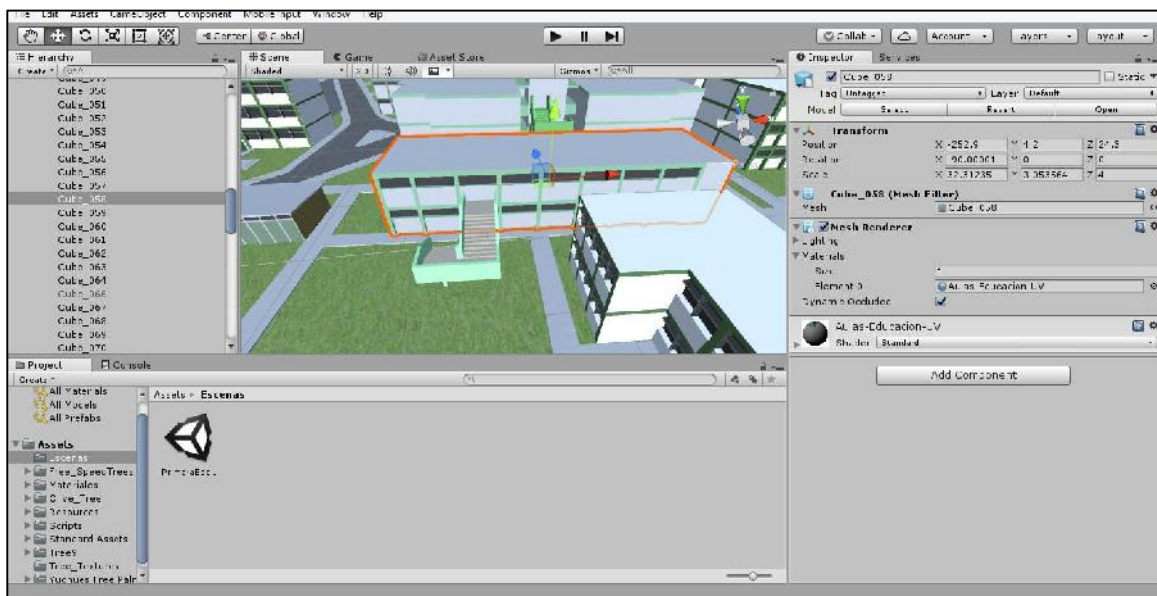


Figura 81: Mesh de las aulas de FACHSE block "1", Objeto 3D
Fuente: Elaboración Propia

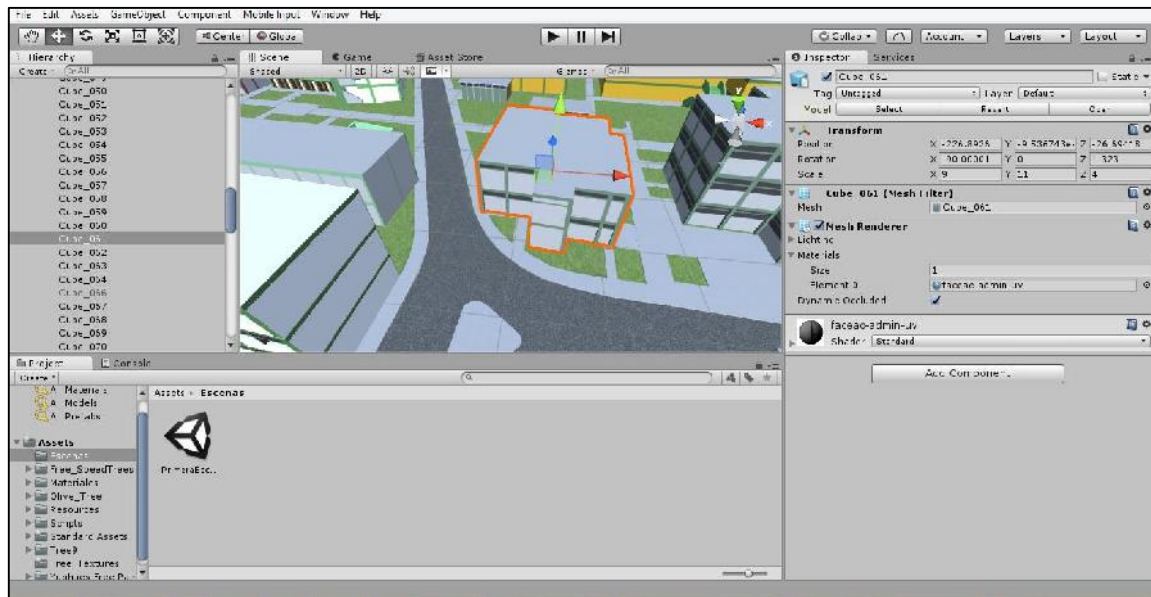


Figura 82: Mesh de FACEAC, Objeto 3D
Fuente: Elaboración Propia

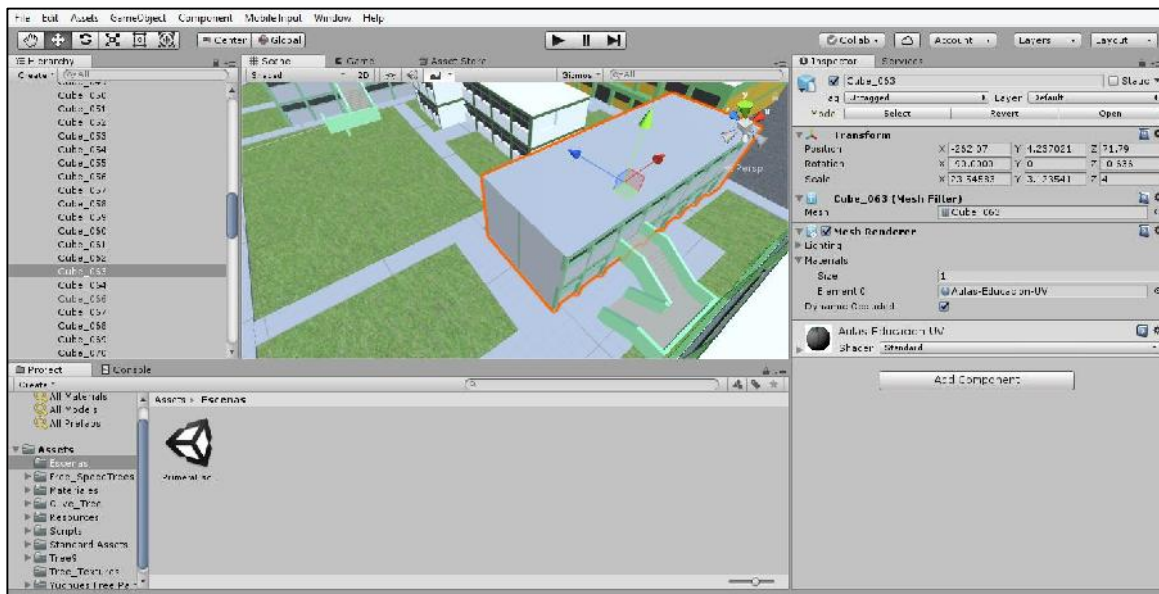


Figura 83: Mesh de las aulas de FACHSE block "2", Objeto 3D
Fuente: Elaboración Propia

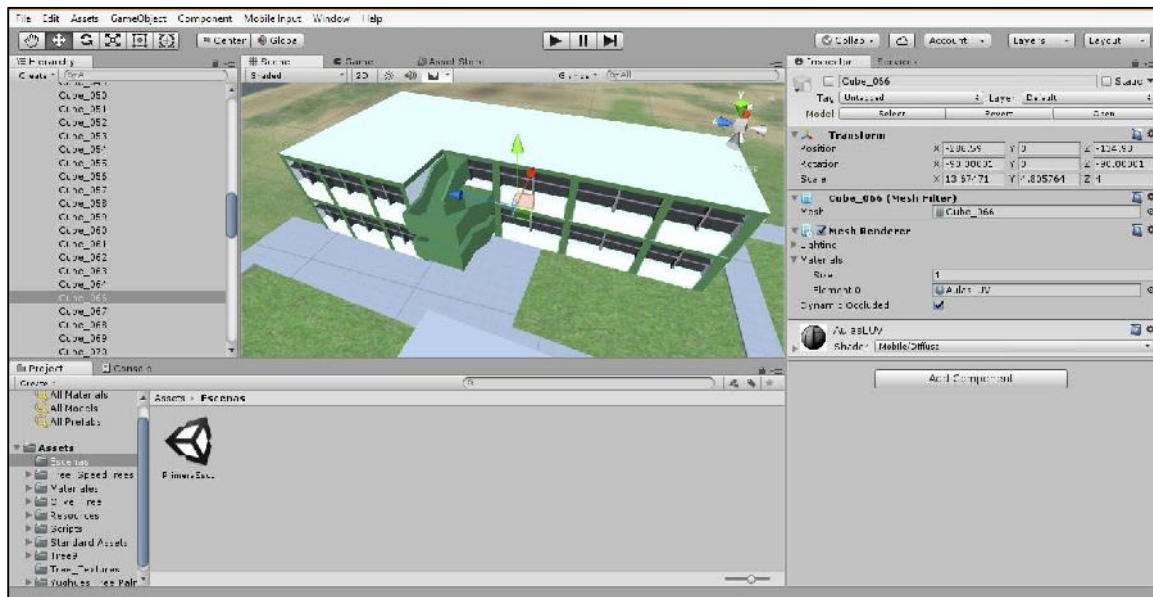


Figura 84: Mesh de FIZ, Objeto 3D
Fuente: Elaboración Propia

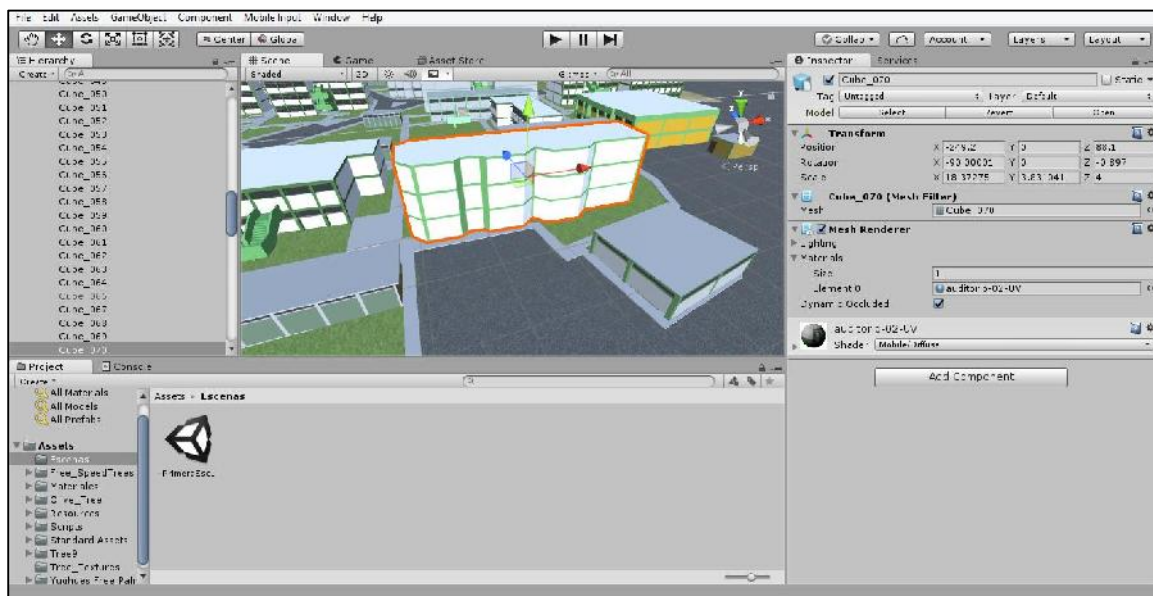


Figura 85: Mesh del Auditorio, Objeto 3D
Fuente: Elaboración Propia

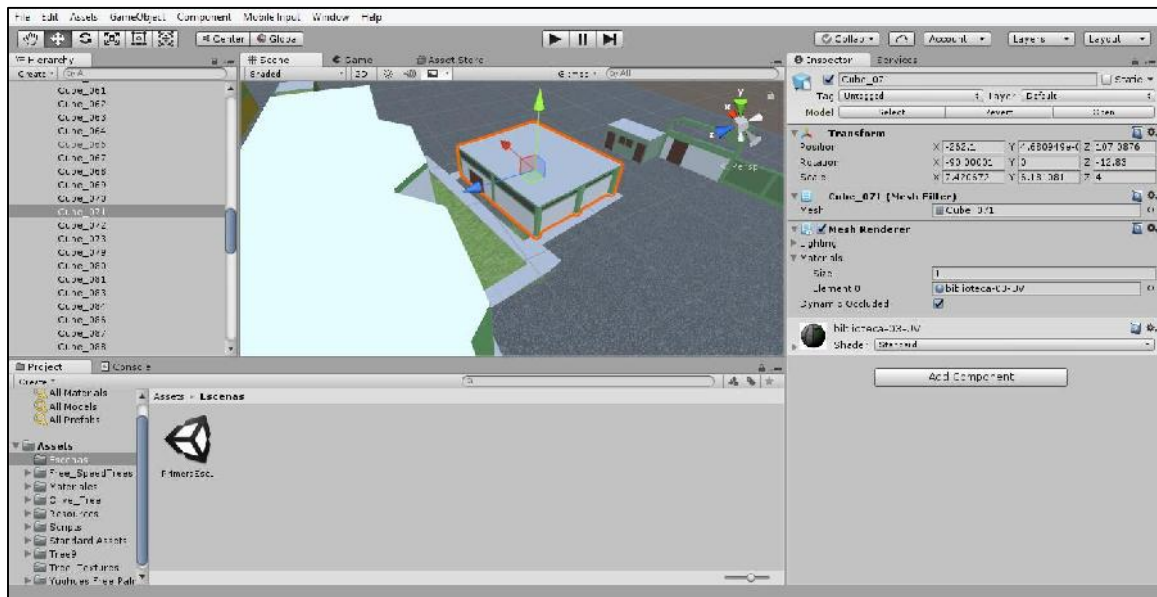


Figura 86: Mesh de la Biblioteca, Objeto 3D
Fuente: Elaboración Propia

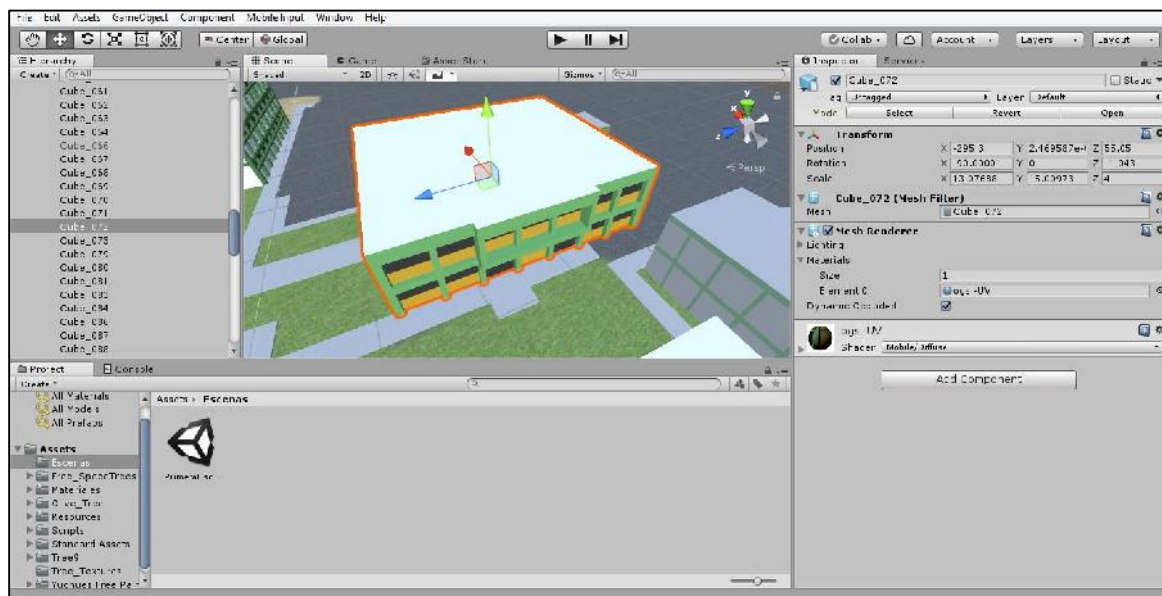


Figura 87: Mesh de la Oficina Central de Asuntos Academicos, Objeto 3D
Fuente: Elaboración Propia

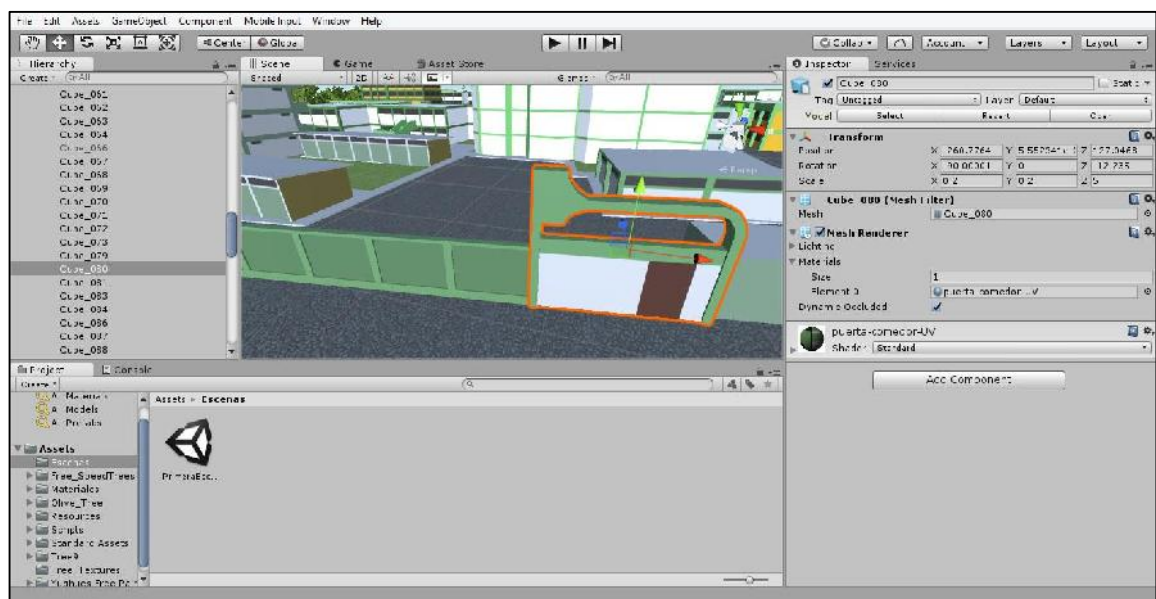


Figura 88: Mesh de la Puerta Sur, Objeto 3D
Fuente: Elaboración Propia



Figura 89: Mesh de la Banda de Concierto, Objeto 3D
Fuente: Elaboración Propia

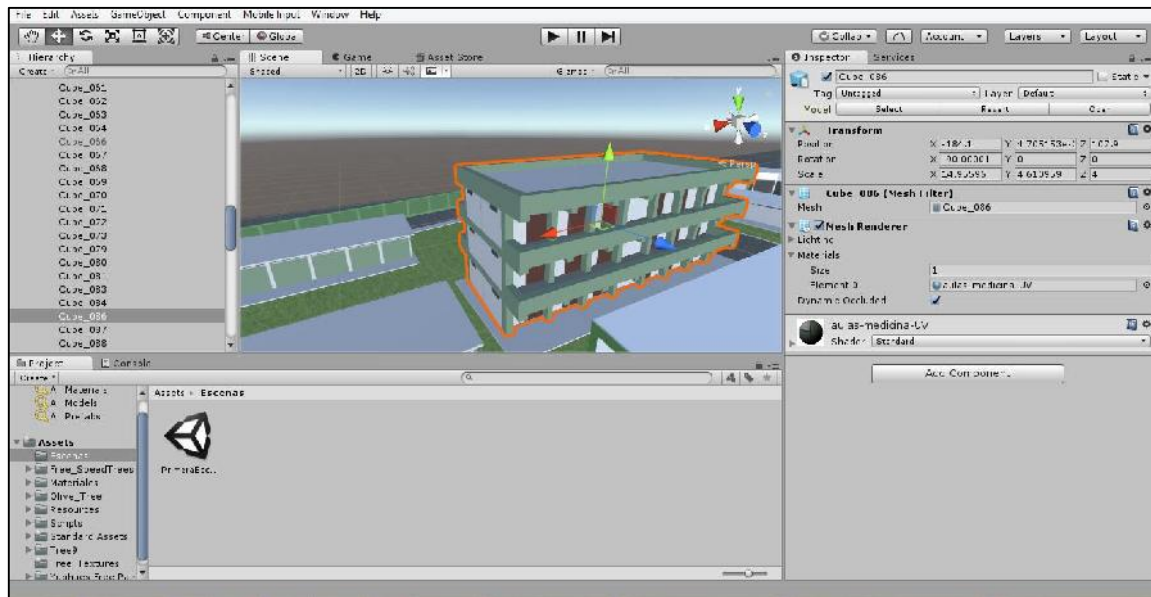


Figura 90: Mesh de las aulas de Medicina Humana, Objeto 3D
Fuente: Elaboración Propia

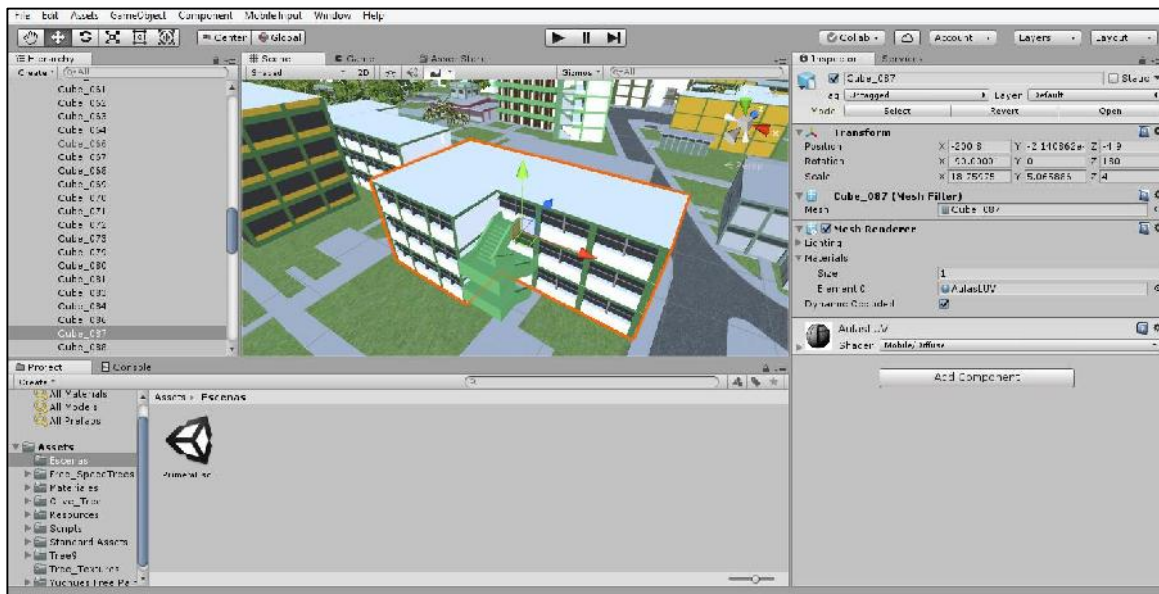


Figura 91: Mesh de la Facultad de Enfermería, Objeto 3D
Fuente: Elaboración Propia

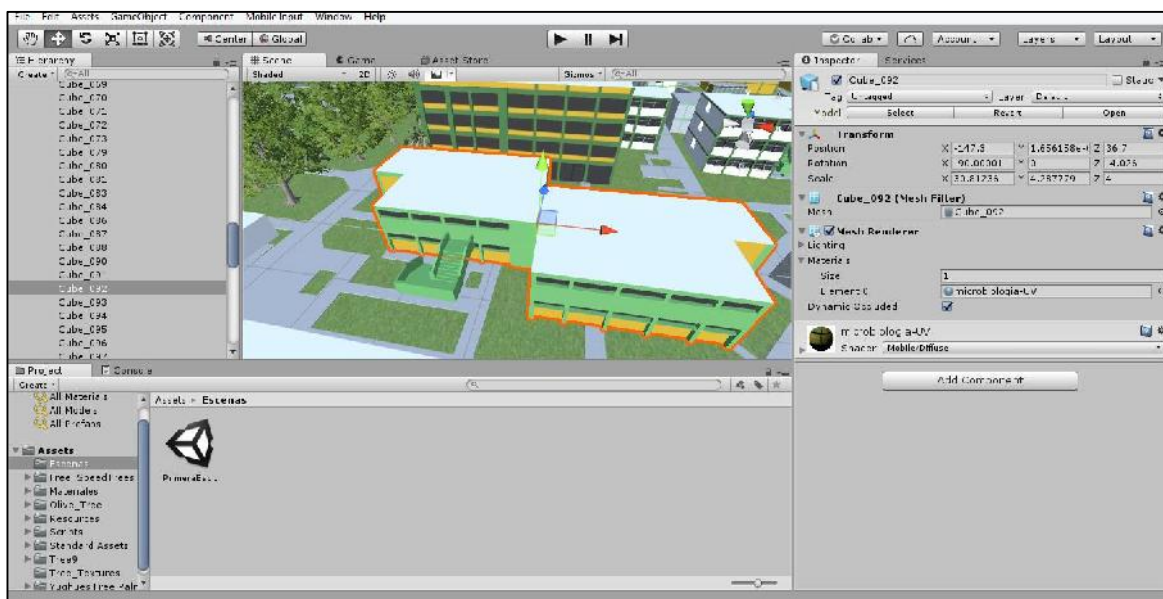


Figura 92: Mesh del Laboratorio de Biología y Microbiología, Objeto 3D
Fuente: Elaboración Propia

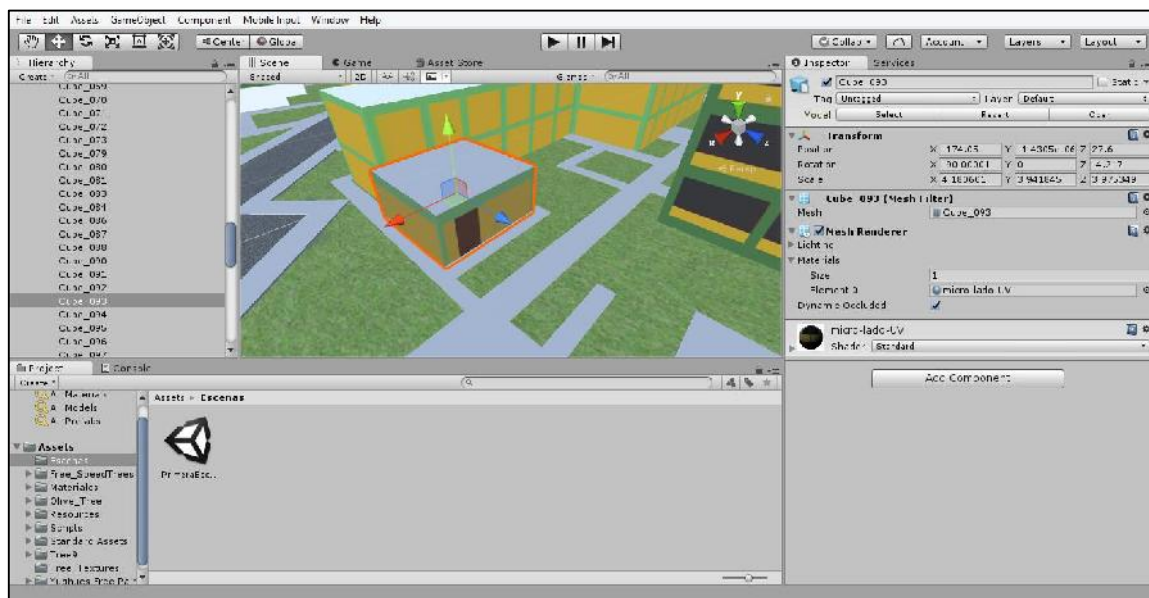


Figura 93: Mesh del Laboratorio de Fitopatología, Objeto 3D
Fuente: Elaboración Propia

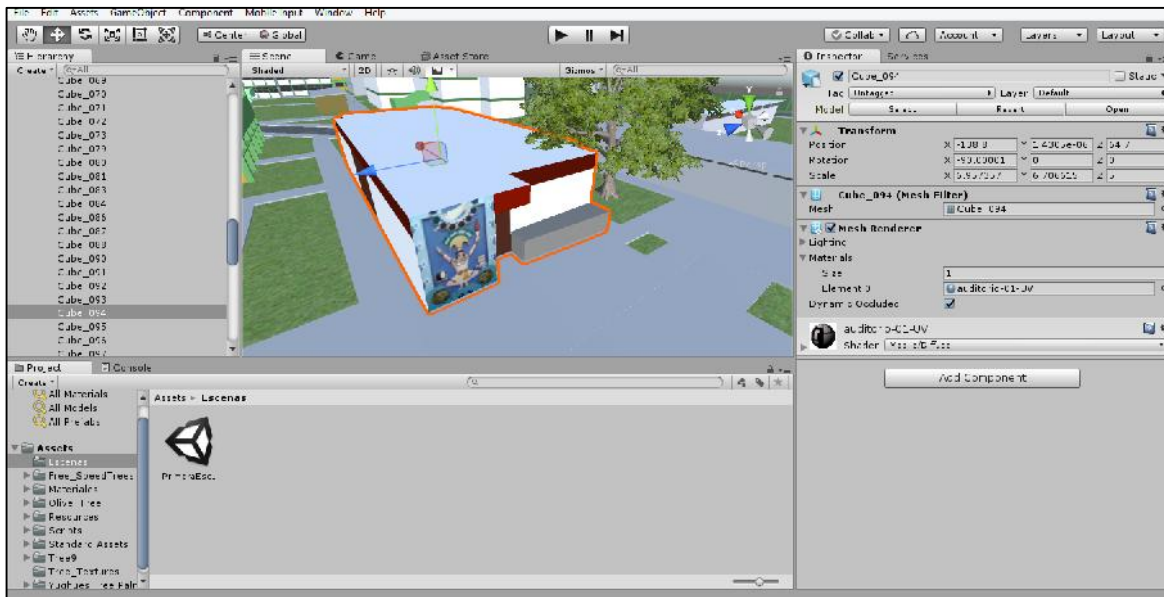


Figura 94: Mesh del Auditorio Principal, Objeto 3D
Fuente: Elaboración Propia

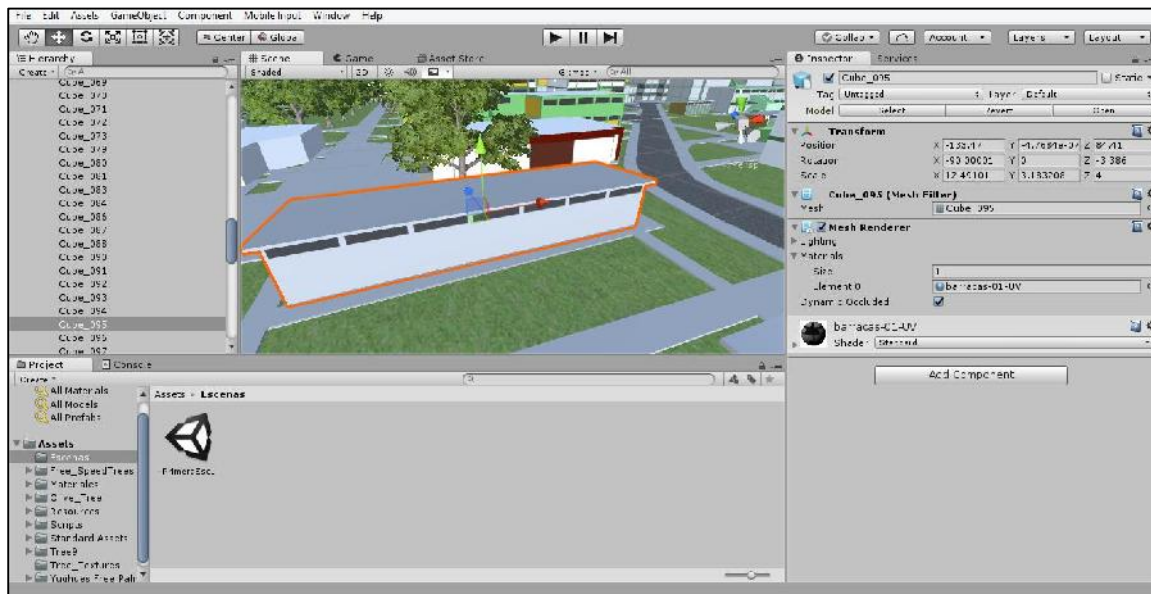


Figura 95: Mesh de la Barracas 1, Objeto 3D
Fuente: Elaboración Propia

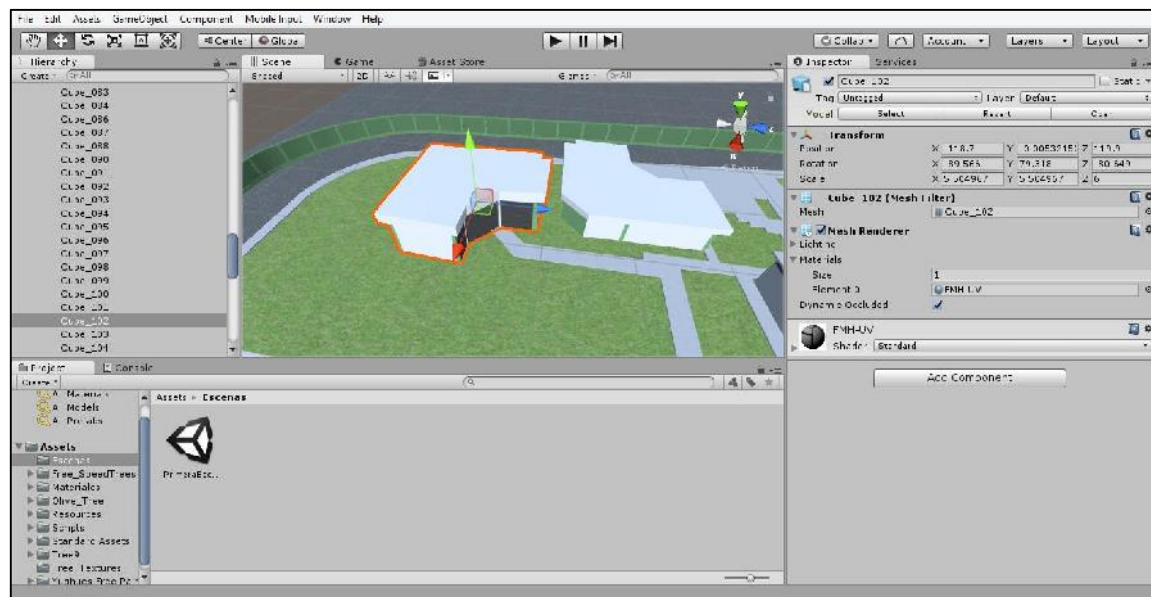


Figura 96: Mesh de la Facultad de Medicina Humana, Objeto 3D
Fuente: Elaboración Propia

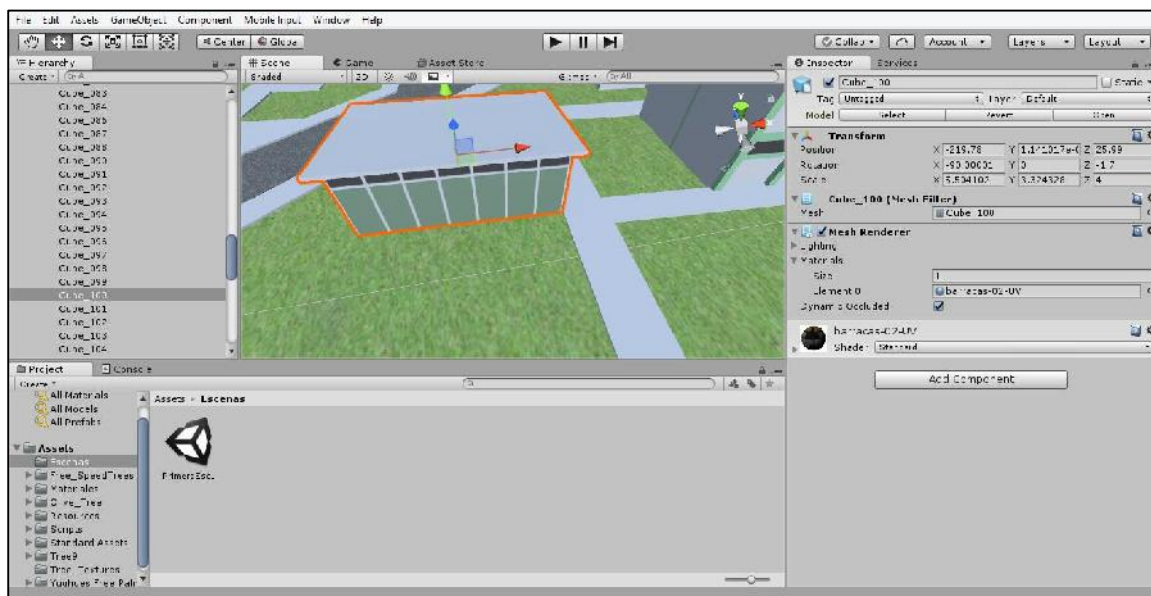


Figura 97: Mesh de la Barracas 2, Objeto 3D
Fuente: Elaboración Propia

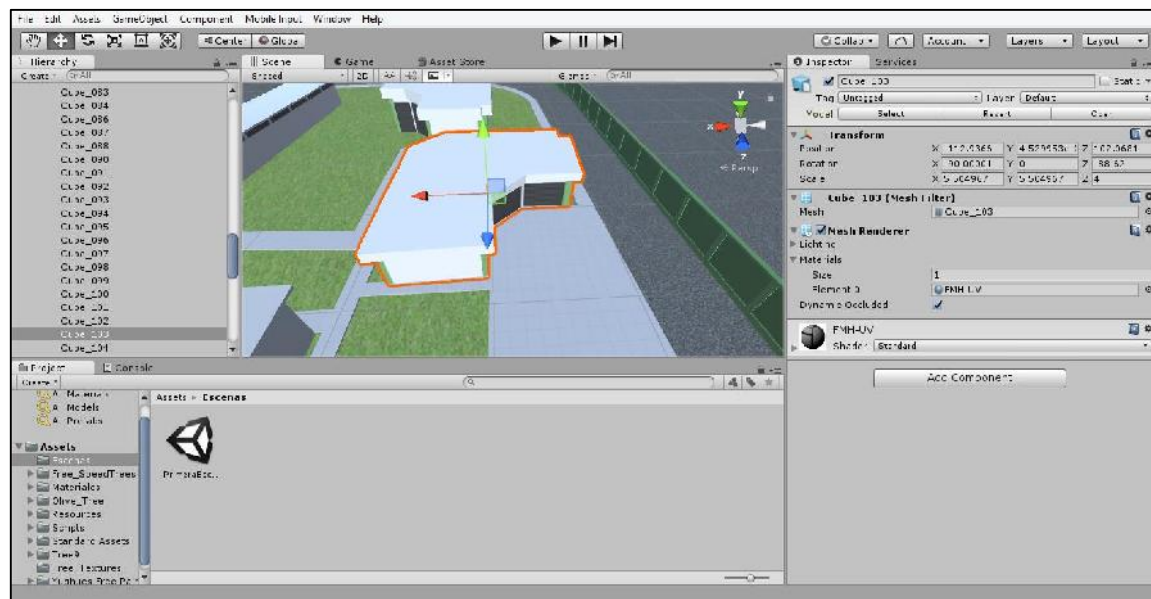


Figura 98: Mesh de Grados y Titulos, Objeto 3D
Fuente: Elaboración Propia

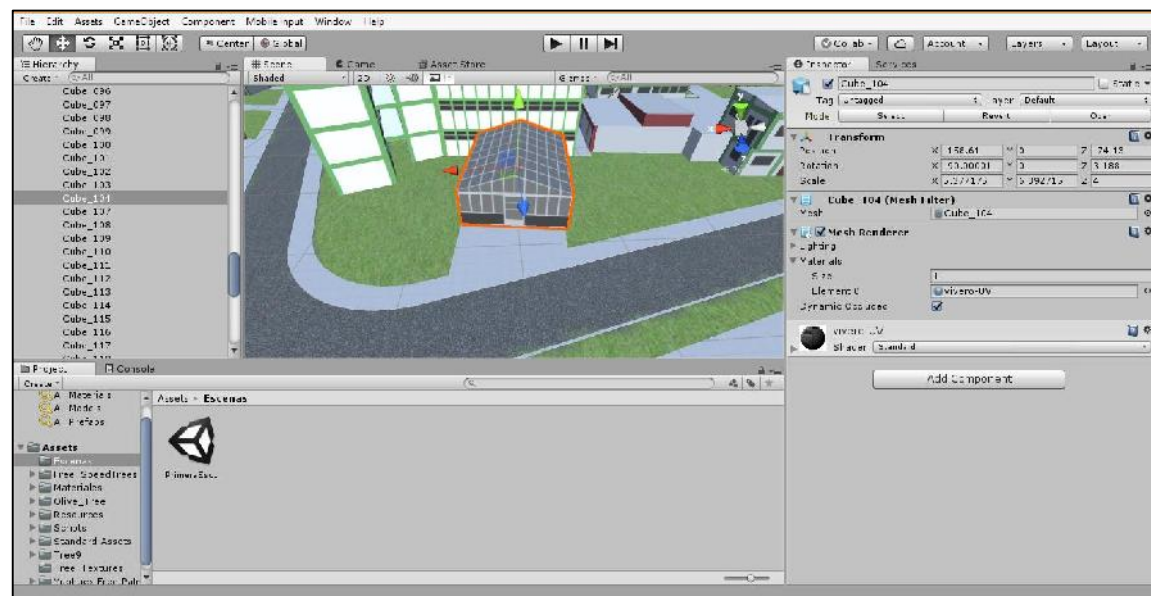


Figura 99: Mesh del Vivero, Objeto 3D
Fuente: Elaboración Propia

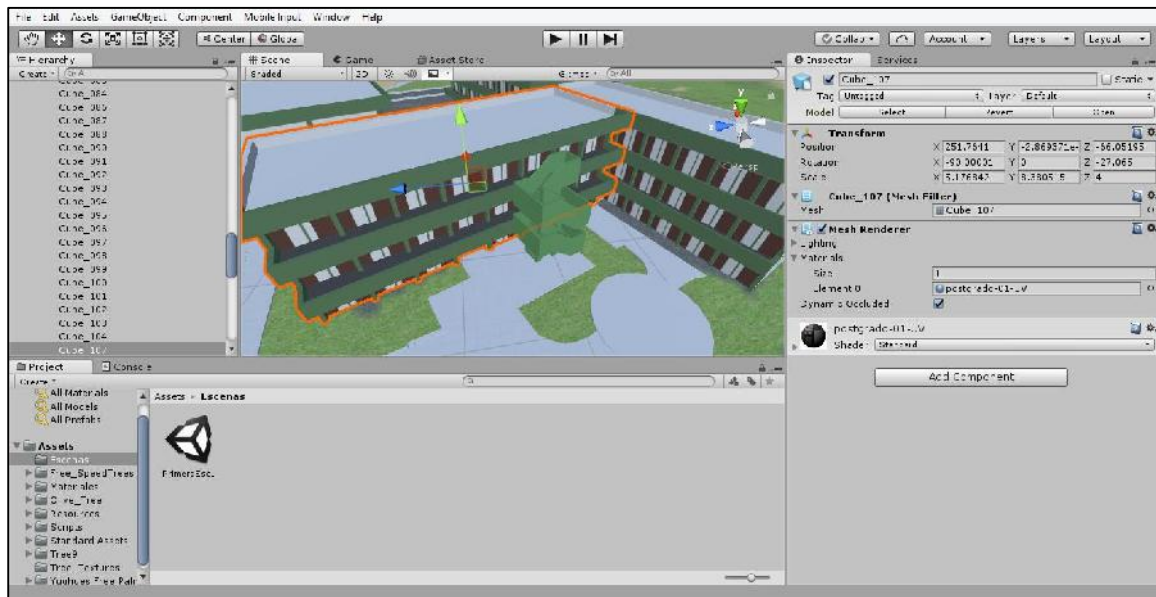


Figura 100: Mesh de las aulas block “B”, Objeto 3D
Fuente: Elaboración Propia

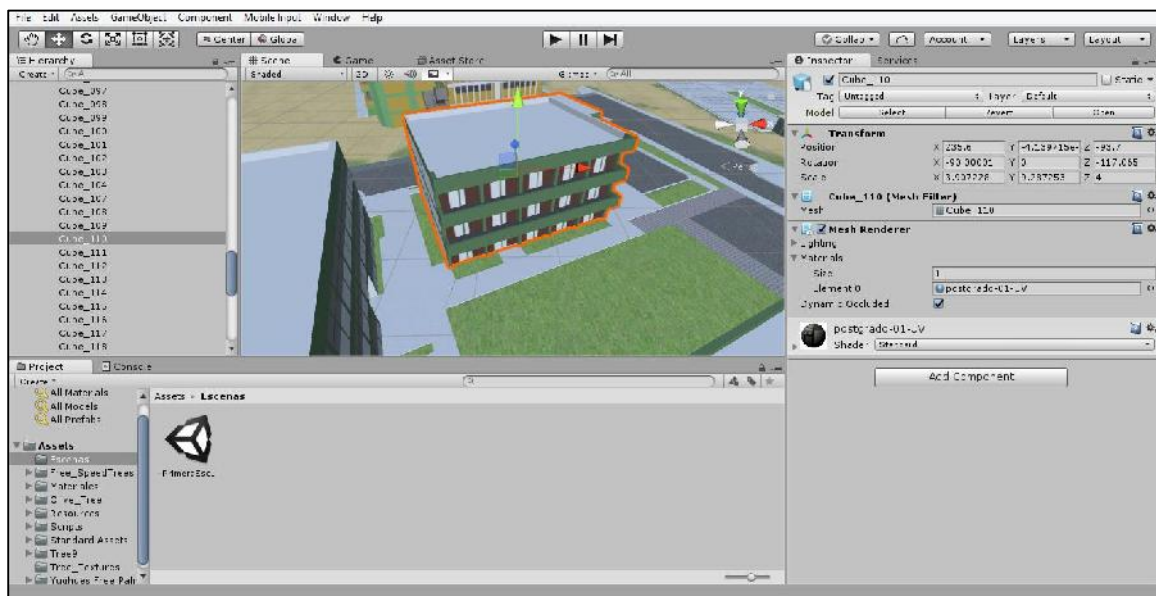


Figura 101: Mesh de las aulas block “C”, Objeto 3D
Fuente: Elaboración Propia

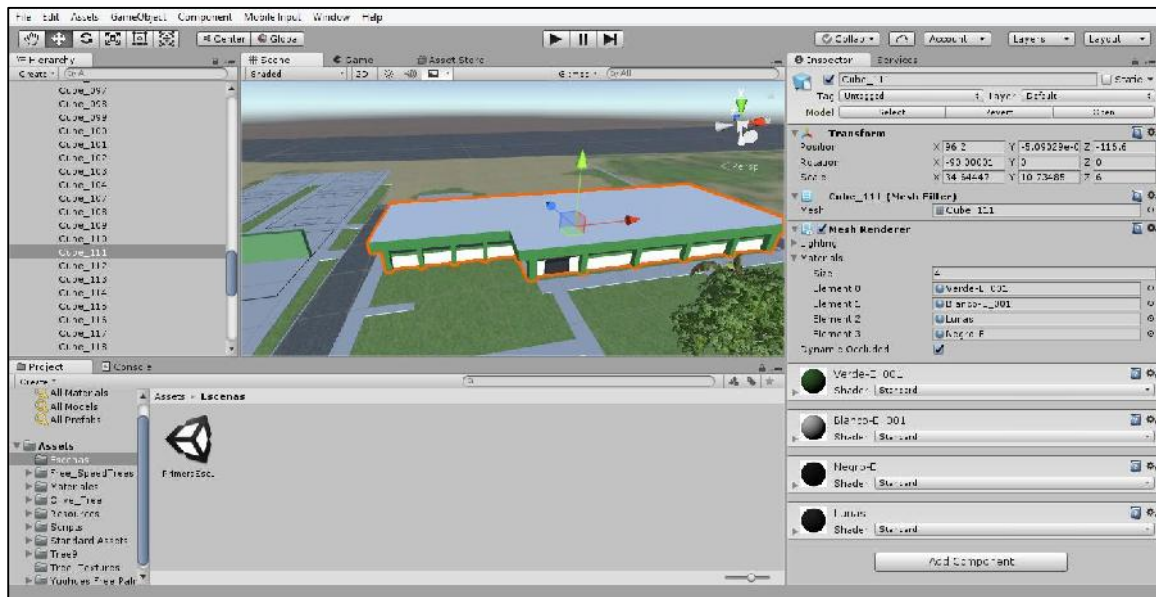


Figura 102: Mesh de las Biblioteca Principal, Objeto 3D
Fuente: Elaboración Propia

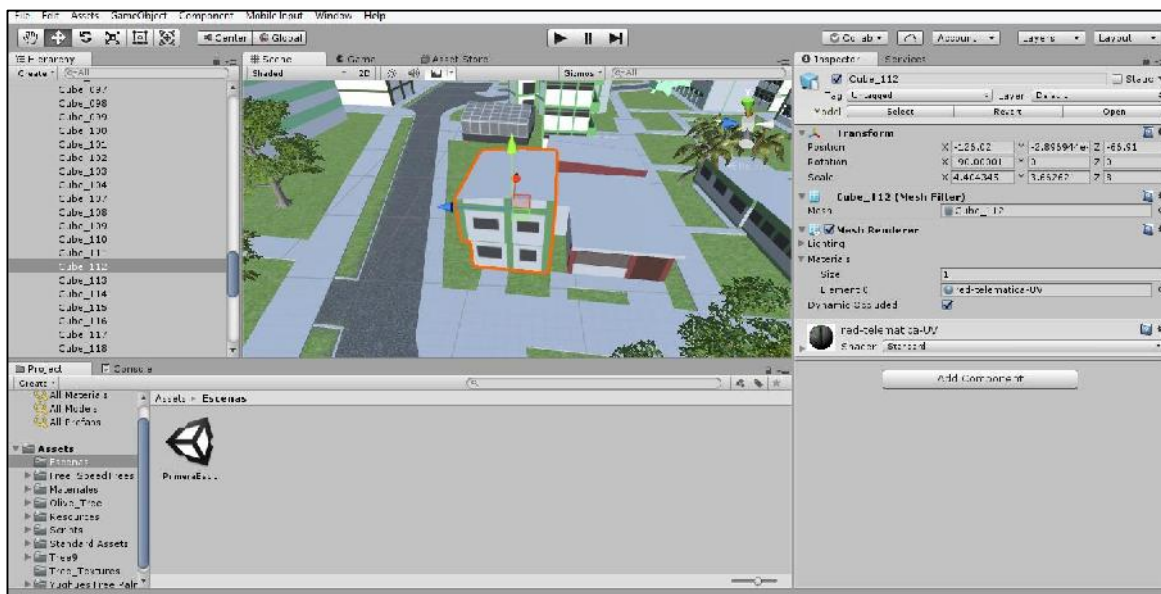


Figura 103: Mesh de la Red Telematica, Objeto 3D
Fuente: Elaboración Propia

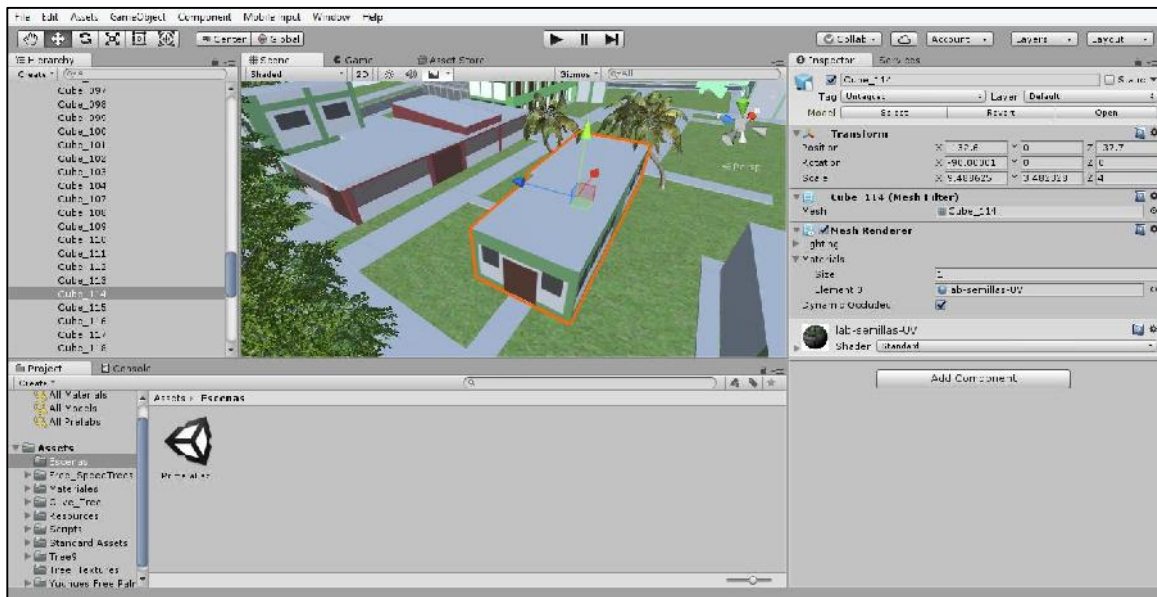


Figura 104: Mesh del Laboratorio de Semillas, Objeto 3D
Fuente: Elaboración Propia

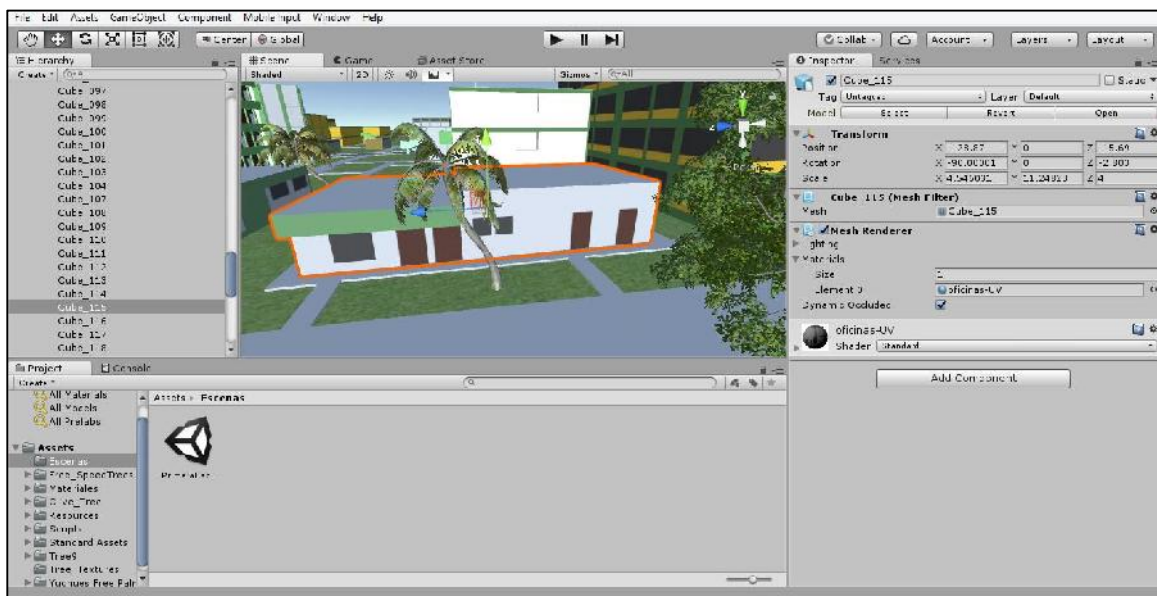


Figura 105: Mesh de la Oficina de Planillas, Objeto 3D
Fuente: Elaboración Propia

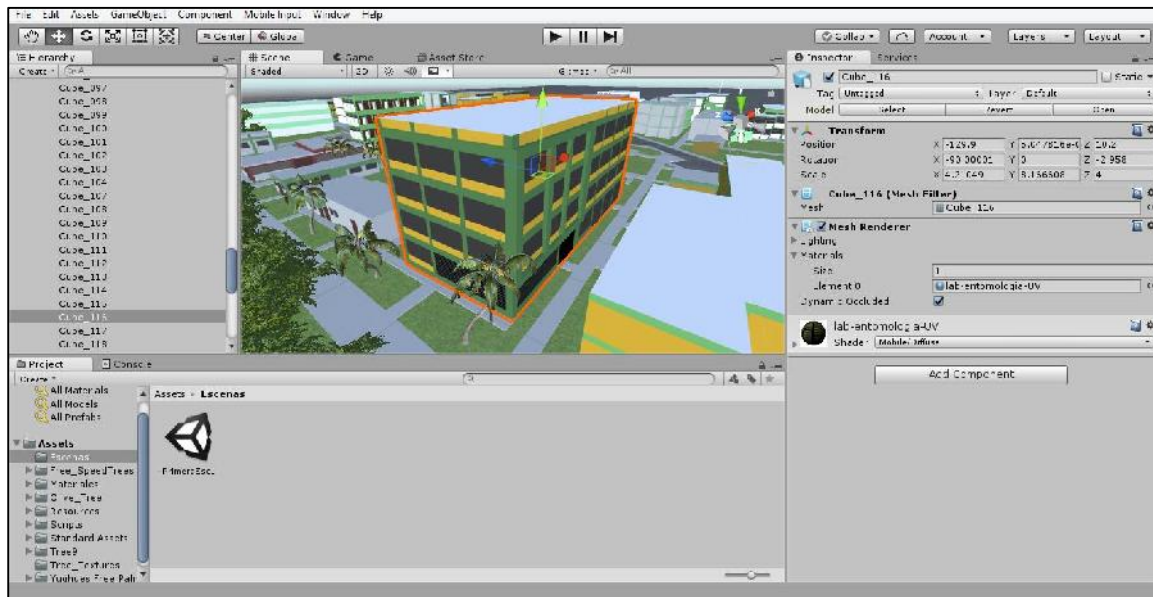


Figura 106: Mesh de la Oficina Central de Investigacion, Objeto 3D
Fuente: Elaboración Propia

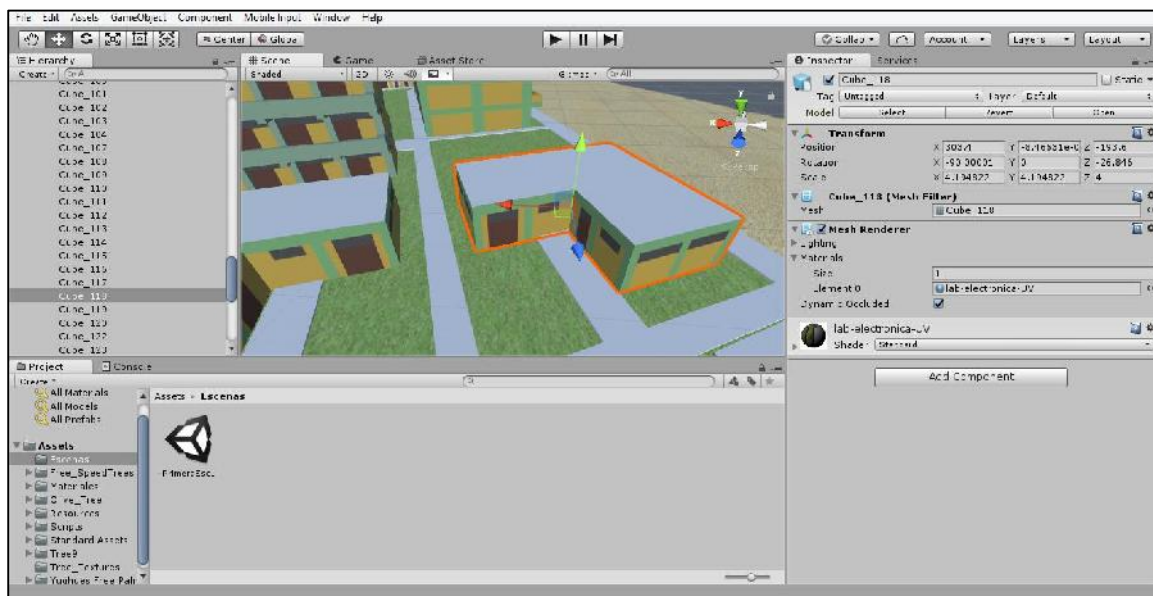


Figura 107: Mesh del Laboratorio de Mecanica, Objeto 3D
Fuente: Elaboración Propia

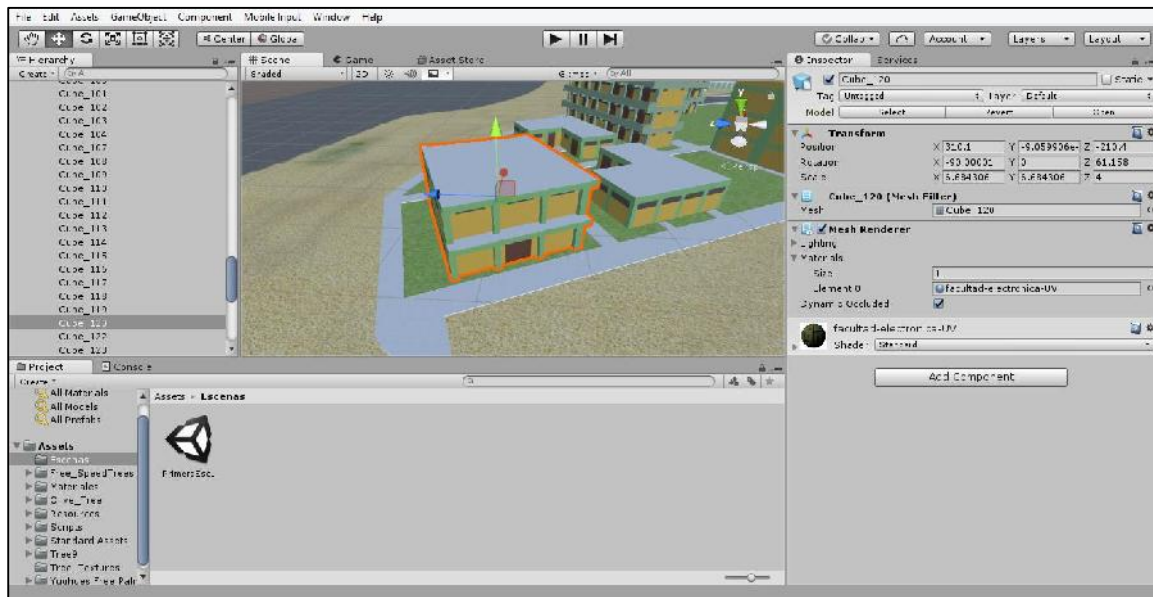


Figura 108: Mesh de FIME, Objeto 3D
Fuente: Elaboración Propia



Figura 109: Mesh del Centro de Producción y Servicios Electromecánicos, Objeto 3D
Fuente: Elaboración Propia

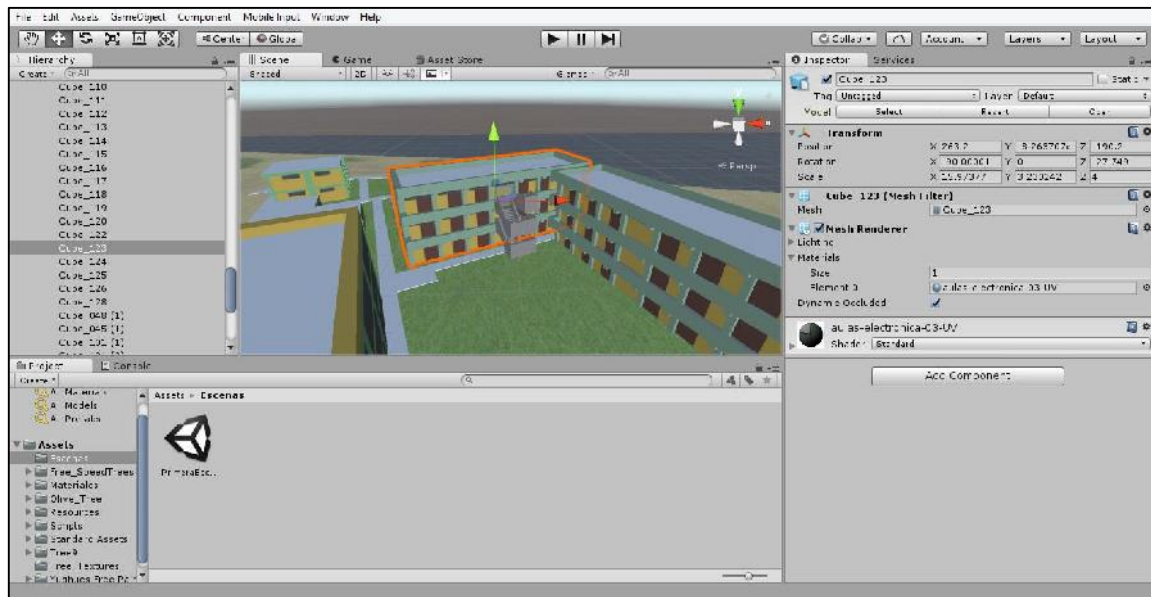


Figura 110: Mesh de las aulas de Mecanica, Objeto 3D
Fuente: Elaboración Propia

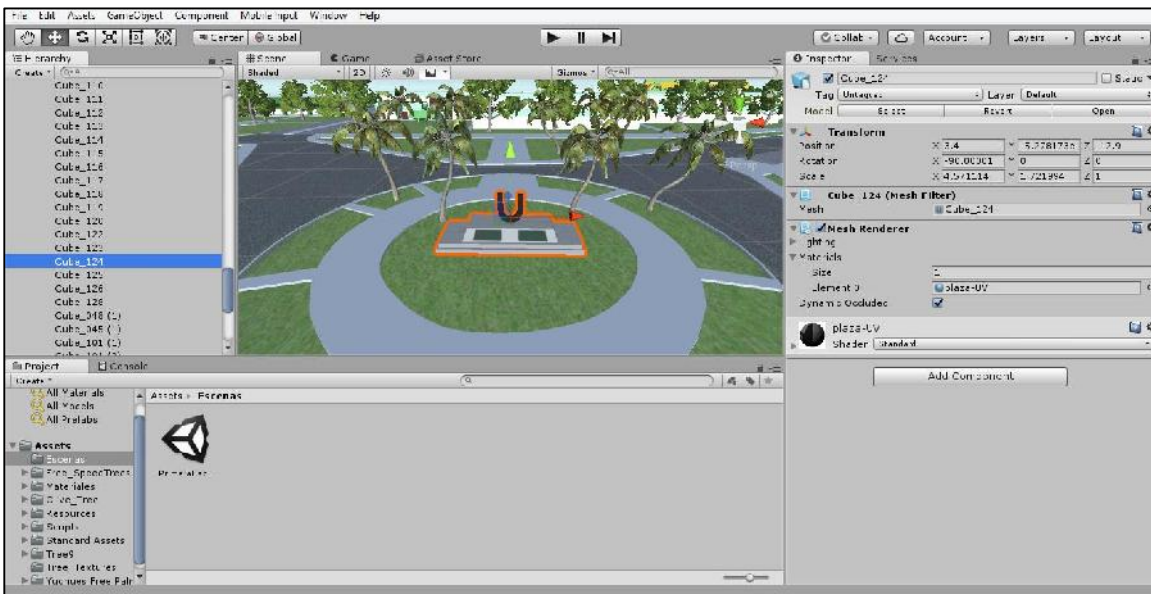


Figura 111: Mesh de la Plaza del Saber, Objeto 3D
Fuente: Elaboración Propia

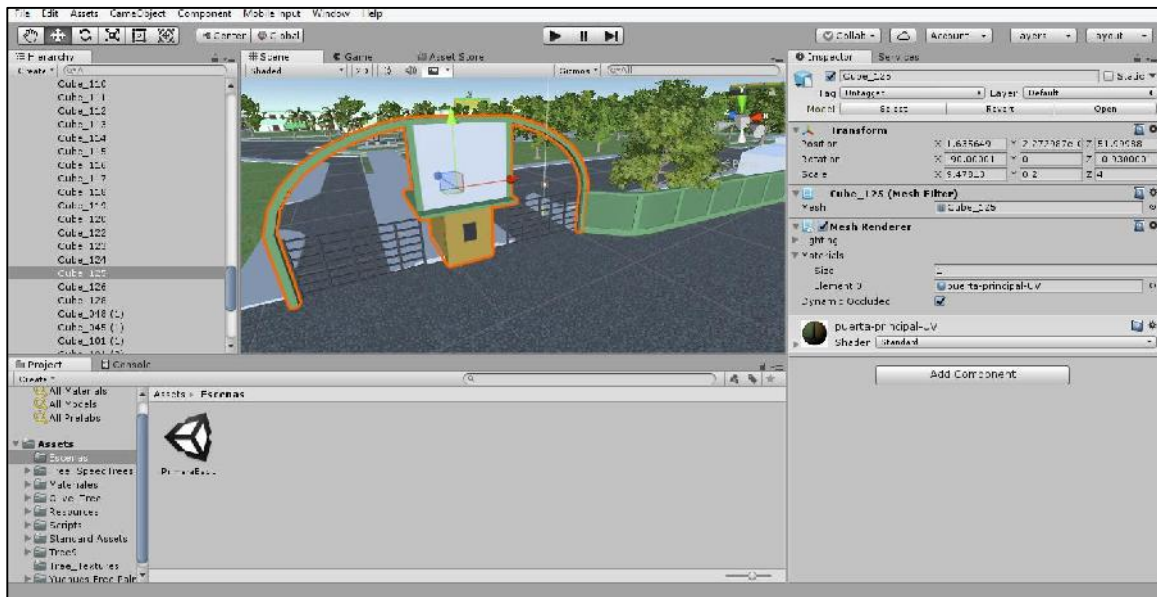


Figura 112: Mesh de la Puerta Principal, Objeto 3D
Fuente: Elaboración Propia

4) Escena

La interfaz grafica de la aplicación móvil está desarrollada en una escena de Unity 3d separada en 2 vistas:

a) Vista 2D:



Figura 113: Aplicación móvil, Vista 2D
Fuente: Elaboración Propia

b) Vista 3D:

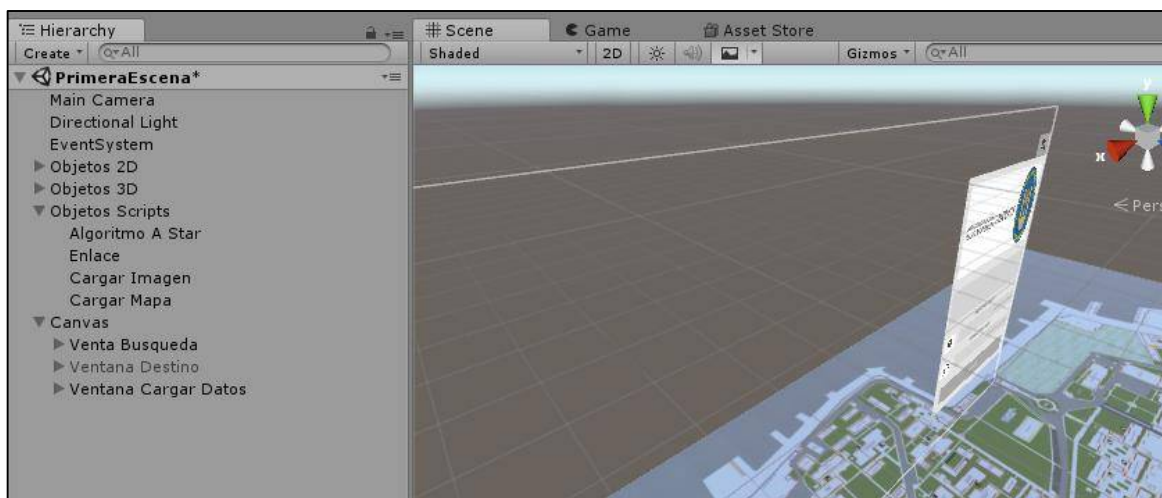


Figura 114: Aplicación móvil, Vista 3D
Fuente: Elaboración Propia

Anexo 8: Encuesta relacionada a la experiencia del usuario con la aplicación móvil en la
Universidad Nacional Pedro Ruiz Gallo

Nombre y Apellidos:.....

1) ¿La aplicación móvil facilitó localizar el destino que buscabas?

- a. Nada
- b. Poco
- c. Regular
- d. Bastante
- e. Mucho

2) ¿Consideras viable la ruta trazada por la aplicación móvil?

- a. Nada
- b. Poco
- c. Regular
- d. Bastante
- e. Mucho

**3) ¿La vista 3-D de la aplicación móvil permitió orientarte en la Universidad Nacional
Pedro Ruiz Gallo?**

- a. Nada
- b. Poco
- c. Regular
- d. Bastante
- e. Mucho

4) ¿Consideras realista el diseño 3-D de la Universidad Nacional Pedro Ruiz Gallo?

- a. Nada
- b. Poco
- c. Regular
- d. Bastante
- e. Mucho

5) ¿Te ha parecido fácil interactuar con la aplicación móvil?

- a. Muy Difícil
- b. Difícil
- c. Regular
- d. Fácil
- e. Muy Fácil

6) ¿Crees que la aplicación móvil es una buena alternativa para localizar las oficinas de la Universidad Nacional Pedro Ruiz Gallo?

- a. Totalmente en desacuerdo
- b. En desacuerdo
- c. Ni de acuerdo ni en desacuerdo
- d. De acuerdo
- e. Totalmente de acuerdo

7) ¿Qué calificación le darías a la aplicación móvil?

- a. Muy mala
- b. Mala
- c. Regular
- d. Buena
- e. Excelente

Tabla 32.
Encuesta N° 2, Respuestas de las personas encuestadas

Encuestados	Preguntas						
Nombres y Apellidos	1	2	3	4	5	6	7
Nunura Paredes Jimmy	Bastante	Bastante	Fácil	Mucho	Bastante	De acuerdo	Buena
Paredes Fernandes Fernando	Bastante	Bastante	Regular	Bastante	Mucho	De acuerdo	Buena
Noriega Quesnay Marzia	Bastante	Bastante	Fácil	Mucho	Mucho	De acuerdo	Buena
Martinez Amaya Sofia	Bastante	Bastante	Regular	Mucho	Mucho	De acuerdo	Buena
Valladolid Silva Raul	Regular	Regular	Difícil	Regular	Regular	Ni de acuerdo ni en desacuerdo	Regular
Ramos Aguirre Vanessa	Mucho	Bastante	Fácil	Bastante	Mucho	De acuerdo	Buena
Sandoval Delgado Dante	Regular	Bastante	Regular	Mucho	Bastante	De acuerdo	Buena
Rojas Torres Robert	Bastante	Bastante	Fácil	Mucho	Mucho	De acuerdo	Buena
Silva Lozada Angeles	Bastante	Bastante	Muy Fácil	Bastante	Mucho	De acuerdo	Buena
Centurion Perales Junior	Bastante	Bastante	Regular	Mucho	Mucho	De acuerdo	Buena
Vasquez Tapia Heber	Bastante	Bastante	Fácil	Mucho	Bastante	De acuerdo	Buena
Fuentes Guevara Erik	Mucho	Mucho	Fácil	Mucho	Mucho	Totalmente de acuerdo	Excelente
Racchumi Iopez Hector	Bastante	Bastante	Regular	Bastante	Mucho	De acuerdo	Buena
Alvarado Chavez Cecilia	Bastante	Regular	Fácil	Regular	Mucho	De acuerdo	Buena
Fuentes Sosa Maria	Bastante	Bastante	Fácil	Mucho	Bastante	De acuerdo	Buena
Rodriguez Chavez Brayan	Bastante	Bastante	Fácil	Mucho	Mucho	De acuerdo	Buena
Siesquen Flores Karen	Bastante	Bastante	Regular	Mucho	Mucho	De acuerdo	Buena
Jacabo polo Ana Maria	Mucho	Mucho	Muy Fácil	Mucho	Bastante	De acuerdo	Buena
Vilchez Chozo Javier	Bastante	Bastante	Regular	Mucho	Mucho	De acuerdo	Buena
Ramos Salvador David	Mucho	Bastante	Fácil	Bastante	Mucho	Totalmente de acuerdo	Excelente

Fuente: Elaboración propia

